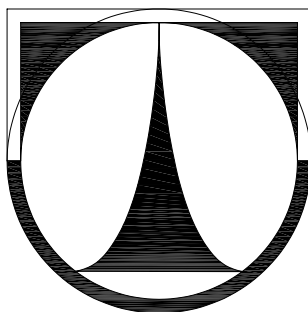


TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta strojní



DIPLOMOVÁ PRÁCE

**Matematický model kinematiky robotizovaného podvozku se
šestnácti stupni volnosti**

**Mathematical Model of Robotized Chassis Kinematics with Sixteen
Degrees of Freedom**

2007

Miroslav Denk

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta strojní

Katedra mechaniky, pružnosti a pevnosti

Studijní program: M2301 - strojní inženýrství

Studijní obor: 3901T003 - Aplikovaná mechanika

Zaměření: Inženýrská mechanika

**Matematický model kinematiky robotizovaného podvozku se
šestnácti stupni volnosti**

**Mathematical Model of Robotized Chassis Kinematics with Sixteen
Degrees of Freedom**

Denk Miroslav

číslo diplomové práce: KMP - 19 266

Vedoucí práce: **Doc. Ing. Miroslav Šír, CSc**

Počet stran:	60
Počet obrázků:	20
Počet grafů:	2
Počet vzorců:	89
Počet příloh:	1

23. května 2007

Anotace

Tato diplomová práce se zabývá tvorbou matematického modelu robotizovaného podvozku se šestnácti stupni volnosti. Tento model bude v budoucnu využit pro stanovení parametrů pohonů, bude součástí řídicího systému a bude základem pro tvorbu simulátoru. Podvozek je opatřen čtyřmi nohami zakončenými koly. Každá noha má čtyři stupně volnosti. Kinematika je řešena v rozsahu úlohy polohy a úlohy rychlosti při maximálním možném dodržení podmínek valení mezi koly a pojezdovou plochou.

Annotation

This diploma thesis deals with elaboration of a mathematical model of a robotized chassis with sixteen degrees of freedom. This model will be used for setting of gear parameters, it will be a part of control system, and it will be a base for creation of a simulator in the future. The chassis is equipped with four shanks ended with wheels. Each shank has four degrees of freedom. The kinematics is solved within the range of location and speed with maximal observance of rolling conditions between the wheels and the surface.

Prohlašuji, že jsem diplomovou práci vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 - školní dílo a § 35 - o výdělečném užití díla k vnitřní potřebě školy.

Beru na vědomí, že Technická univerzita v Liberci (TUL) má právo na uzavření licenční smlouvy o užití mé práce a prohlašuji, že souhlasím s případným užitím mé práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Místo: Liberec

Datum: 10.května 2007

Podpis:

Poděkování

Rád bych touto cestou poděkoval všem, kteří mi s vypracováním diplomové práce pomohli.

Zejména bych chtěl poděkovat svému vedoucímu diplomové práce Doc. Ing. Miroslavu Šírovi, CSc z Katedry mechaniky, pružnosti a pevnosti, který mi poskytnul svůj čas, odborný dohled a mnohé cenné rady.

Obsah

Anotace	4
Annotation	4
Obsah	7
Seznam použitých symbolů	9
Úvod	11
Rešerše	11
Nabízená řešení pro pohyb handicapovaných osob v otevřeném terénu	11
Volba koncepce	14
Parametry robotizovaného podvozku	16
Zaměření práce	17
1 Použité matematické postupy	18
1.1 Maticová metoda v kinematice	18
1.1.1 Kinematika tělesa ve 3D	18
1.1.2 Rozšířené matice	21
1.1.2.1 Současné pohyby	23
1.2 Určení normály k ploše	24
1.3 Určení vzdálenosti bodu od přímky	25
1.4 Numerické řešení soustav diferenciálních rovnic	26
1.4.1 Runge - Kuttova metoda 4. řádu	26
1.5 Software Maple	27
2 Souřadnicové systémy	28
2.1 Globální souřadnicový systém GSS	29
2.2 Souřadnicový systém (L, x^L, y^L, z^L)	29
2.2.1 Systém sférických úhlů RPY	30
2.3 Souřadnicové systémy $_i(U, x^U, y^U, z^U)$, $i=1,2,\dots,4$	31
2.4 Souřadnicové systémy $_i(A, x^A, y^A, z^A)$, $i=1,2,\dots,4$	32
2.5 Souřadnicové systémy $_i(B, x^B, y^B, z^B)$, $i=1,2,\dots,4$	32
2.6 Souřadnicové systémy $_i(C, x^C, y^C, z^C)$, $i=1,2,\dots,4$	33

2.7	Souřadnicové systémy $_i(D, x^D, y^D, z^D)$, $i=1,2,\dots,4$	34
2.8	Souřadnicové systémy $_i(E, x^E, y^E, z^E)$, $i=1,2,\dots,4$	35
2.9	Body dotyku i -tého kola s pojezdovou plochou $f(x,y)$	36
3	Vlastní matematický model.....	37
3.1	Úloha polohy	38
3.2	Úloha rychlosti	41
3.2.1	Derivace soustavy (3.6) podle času.....	41
3.2.2	Podmínky valení	42
3.2.2.1	Rychlost dotykového bodu považovaného za bod i -tého kola.....	42
3.2.2.2	Přiřazení podmínek valení.....	43
3.2.2.3	Podmínka valení u dokonale se valícího kola.....	44
3.2.2.4	Podmínka valení u kola, které má definovanou podmínku valení v daném směru...	45
3.3	Vlastní řešení soustavy	46
3.3.1	Počáteční podmínky pro soustavu (3.23)	47
4	Základní manévry.....	48
4.1	Jízda do zatáčky	48
4.2	Průjezd zúženým místem	52
4.3	Překonání překážky překročením.....	55
4.4	Pohyb do schodů	57
5	Závěr	58
6	Seznam použité literatury	59
7	Seznam příloh.....	60

Seznam použitých symbolů

Označení:	Jednotka:	Název veličiny:
A, B, C, D, E, G, L, U	[-]	počátky souřadnicových systémů
${}_i(A, x^A, y^A, z^A)$	[-]	označení souřadnicového systému s počátkem v bodě ${}_iA$ s osami x^A, y^A, z^A příslušející i -té noze
\mathbf{r}_L^G	[m]	polohový vektor bodu L vyjádřený v GSS
${}_i\mathbf{r}_U^L$	[m]	polohový vektor bodu ${}_iU$ vyjádřený v souřadnicovém systému L, x^L, y^L, z^L příslušný i -té noze
$\bar{\mathbf{r}}_L^G$	[m]	rozšířený tvar vektoru \mathbf{r}_L^G
$\bar{\mathbf{v}}_D^G$	[m/s]	rychlost bodu D vyjádřená GSS
$(\bar{\mathbf{v}}_T^G)^E$	[m/s]	označení rychlosti bodu T v GSS, jako by byl součástí souřadnicového systému ${}_i(E, x^E, y^E, z^E)$
$(\bar{\mathbf{v}}_T^G)_x^E$	[m/s]	x -ová složka vektoru $(\bar{\mathbf{v}}_T^G)^E$
$\mathbf{T}^1, \mathbf{T}^2, \mathbf{T}^3$	[-]	transformační matice, které reprezentují otočení okolo příslušných os souřadnicového systému
\mathbf{T}^{GL}	[-]	transformační matice pro transformaci ze souřadnicového systému L, x^L, y^L, z^L do GSS
${}_i\mathbf{T}^{LU}$	[-]	transformační matice pro transformaci ze souřadnicového systému ${}_i(U, x^U, y^U, z^U)$ do (L, x^L, y^L, z^L) příslušná i -té noze
$\bar{\mathbf{T}}^{GL}$	[-]	rozšířený tvar matice \mathbf{T}^{GL}
${}_i\mathbf{n}^{0G}$	[-]	jednotkový normálový vektor k ploše $f(x, y)$ s počátkem v bodě ${}_iT$
${}_i\mathbf{v}_T^G$	[m/s]	označení rychlosti bodu T v globálním souřadnicovém systému
$ \mathbf{a} $	[-]	norma vektoru \mathbf{a}
$\mathbf{a} \cdot \mathbf{b}$	[-]	skalární součin vektorů \mathbf{a}, \mathbf{b}

$\mathbf{a} \times \mathbf{b}$	[-]	vektorový součin vektorů \mathbf{a} , \mathbf{b}
$i=1,2,\dots,4$	[-]	tento index rozlišuje jednotlivé nohy v pořadí: levá přední, pravá přední, levá zadní, pravá zadní
${}_i\varphi_{Az}$	[rad]	natočení okolo osy z^A i -té nohy
${}_iT$	[-]	bod dotyku i -tého kola s plochou $f(x,y)$
$f(x,y)$	[-]	pojezdová plocha
r	[m]	poloměr kola
R	[m]	poloměr požadované zatáčky
v	[m/s]	požadovaná rychlost průjezdu zatáčkou
φ	[rad]	polohový parametr průjezdu zatáčkou
$p(R)$	[-]	obecná funkce parametru R
${}_id$	[m]	poloměr, na kterém se otáčí i -té kolo

Úvod

Tato diplomová práce vznikla v rámci výzkumného záměru *Optimalizace vlastností strojů v interakci s pracovními procesy a člověkem* a zabývá se robotizovaným podvozkem vozíku pro sociálně zdravotní aplikace s cílem přispět k vývoji zařízení, které umožní pohyb handicapovaných osob a ležících pacientů v obtížném terénu. Řešení úlohy je rozděleno do dvou diplomových prací, jedna se zabývá matematickým modelem („Matematický model kinematiky robotizovaného podvozku se šestnácti stupni volnosti“) a druhá se zabývá konstrukcí základní podvozkové skupiny - kombinované podvozkové nohy („Konstrukce podvozkové nohy robotizovaného podvozku“).

Rešerše

Základním problémem je samotná koncepce podvozku. Proto vlastnímu návrhu předcházela rešeršní činnost s cílem najít analogická řešení presentovaná v otevřených informačních zdrojích.

Invalidních vozíků určených do terénu je možno nalézt celou řadu. Žádný z nich však nemá uspokojivě řešení stabilizaci prostoru pro uživatele a průchodnost a manévrovatelnost terénem řeší spíše hrubou silou, jak ukazují dále uvedené příklady.

Nabízená řešení pro pohyb handicapovaných osob v otevřeném terénu

Jedním z možných řešení je klasická terénní čtyřkolka (ATV - All Terrain Vehicle). V současné době se nabízejí stovky typů od desítek výrobců. Koncepčně jsou však tato vozidla prakticky identická. Mají spalovací motor a náhon 4x4 s rozvodem hnacího momentu prostřednictvím uzamykatelných nápravových a mezinápravových diferenciálů. Jednotlivé cenové kategorie se přitom liší mírou automatizace ovládání těchto diferenciálů. Přední nápravy jsou u většiny typů provedeny jako dvě nezávislá lichoběžníková zavěšení se zvýšeným zdvihem a zadní nápravy bývají většinou tuhé, zavěšené na zkrutných ramenech.

Z hlediska našeho záměru, je základním problémem skutečnost, že vozidla ATV nejsou primárně určena pro handicapované osoby. Částečně se používají pro hospodářské účely, většinou jsou to však prostředky pro provozování „adrenalinových“ sportů. Navíc, zejména v Evropě, nemají povolen přístup do většiny turisticky zajímavých a ekologicky chráněných oblastí. A také zdaleka neřeší veškeré potřeby pohybu handicapovaných osob, například v urbanizovaných pěších územích, kde se běžně vyskytují překážky ve formě schodů, obrubníků chodníků a zúžených profilů.

Mezi další řešení určená přímo pro pohyb handicapovaných osob v terénu patří:

- vozidlo SuperFour od firmy OttoBock (<http://www.ottobock.com>)



Pohon tohoto vozidla je řešen čtyřmi nezávisle elektricky poháněnými koly, ale systém náprav žádné mimořádné řešení nevykazuje. Jedná se o čtyři klasická nezávislá lichoběžníková zavěšení, pouze zdvih je výrazně zvětšen. Co se týče vodorovné stabilizace prostoru pro cestujícího, je zde možnost při sjezdu nebo výjezdu kopce naklonit sedačku, což je pro pohyb v terénu nedostačující. Maximální rychlost vozítka je cca 15 km/h.

- pojízdné křeslo Tankchair (<http://www.tankchair.com/gallery.htm>)



Parametry tohoto vozidla nejsou na webových stránkách uvedeny, ale z obrázku je patrné, že toto řešení je vhodné jen pro venkovní použití, protože díky svým robustním rozměrům není vozík schopen projet úzkým místem, jako jsou např. zárubně dveří. Což vylučuje jeho použití v bytě. A navíc zde není vůbec řešena vodorovná stabilita sedadla.

- 6x6 Explorer (<http://www.kemcare.co.nz>)



Uváděná maximální rychlost vozidla je 4,5-6,5 km/h. Toto řešení také nemá řešení vodorovnou stabilitu sedadla a navíc pevné uložení kol neposkytuje dostatek komfortu při jízdě v terénu.

- Predator 4x4 (<http://www.radicalmobility.com/products.html>)



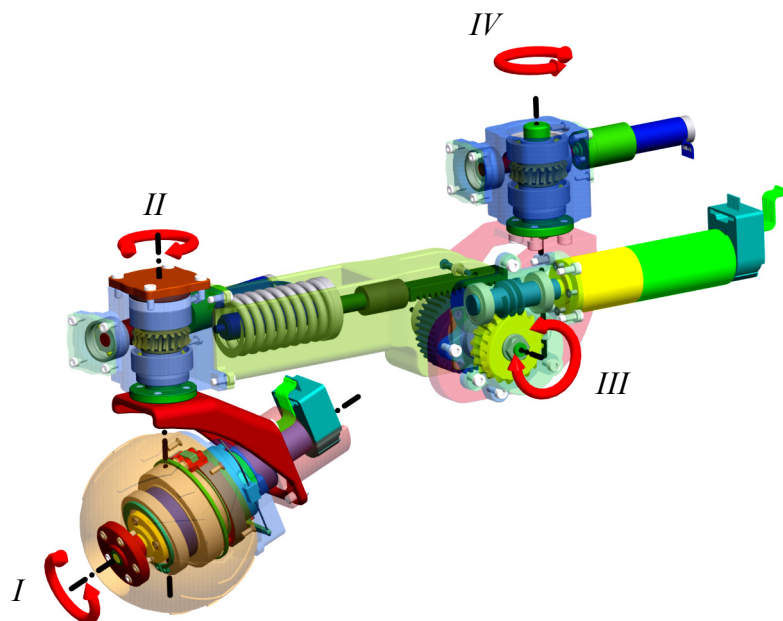
Vozík je poháněn čtyřmi elektromotory, každý o výkonu 250W. Dosahuje rychlosti 7-8km/h. Toto řešení představuje na první pohled klasický elektrický vozík, pouze má větší kola a pohon s vyšším výkonem.

Volba koncepce

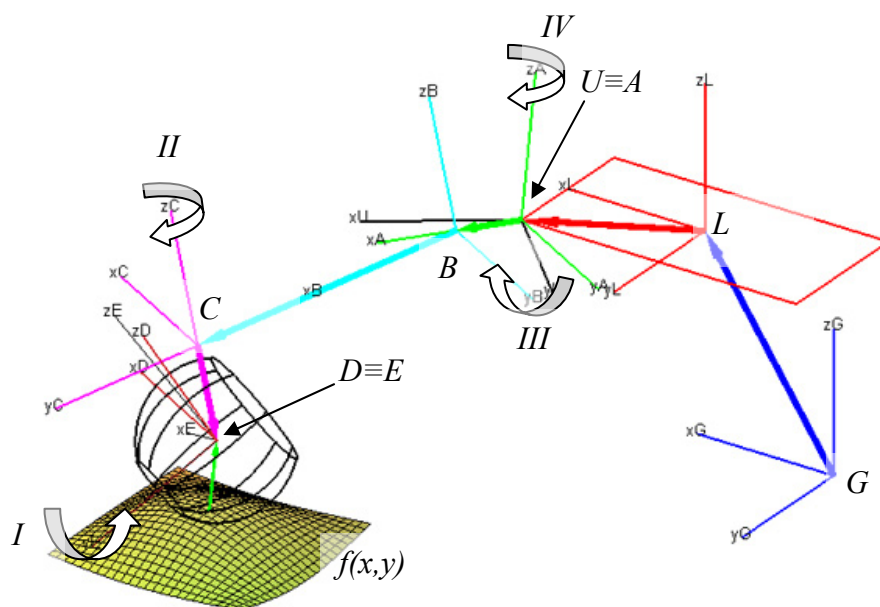
Aby se uživatel mohl volně pohybovat v urbanizovaném prostředí i ve volné přírodě bez pomoci jiné osoby, měla by koncepce podvozku být taková, aby robotizovaný podvozek byl schopen alespoň těchto manévřů při zachování sedačky ve vodorovné poloze:

- jízda v přímém i proměnném směru po rovném i zvlněném terénu,
- změna světlé výšky podvozku,
- pohyb po schodištích různých parametrů,
- překonání překážky překročením,
- průjezd úzkým profilem bez ztráty stability.

Proto byla zvolena konfigurace se čtyřmi nohami, z nichž každá je opatřena kolem. Dále v textu při označení nohy s kolem, bude používán termín noha. Kolo bylo zvoleno kulového tvaru, protože takové kolo je schopné i při větším odklonu osy rotace od tečné roviny jízdy. Každá noha má čtyři stupně volnosti, které jsou přímo nebo nepřímo ovládány samostatnými elektromotory a jsou vyznačeny na následujících obrázcích.



Obr.1: Stupně volnosti nohy vyznačené na konstrukčním řešení¹



Obr.2: Stupně volnosti nohy vyznačené na matematickém modelu

- I. rotace kola
- II. pivotace kola
- III. vyrovnávání terénu
- IV. rejd

¹ Tento obrázek je zde zveřejněn se svolením Jaroslava Korfa, který diplomovou práci týkající se konstrukce podvozkové nohy vypracoval.

Rotací kola se uvádí celé vozidlo do pohybu. Změnou úhlu pivotace se dosahuje změny směru jízdy. Stupeň volnosti označený jako rejď (IV) slouží ke změně rozvoru a rozchodu kol a bude také využíván při překonávání překážek. Stupeň volnosti na Obr.1 a Obr.2 vyznačený jako *III* je úhel, který umožňuje vyrovnávání nerovností terénu a změnu světlé výšky podvozku. Ve fyzické realizaci bude ovládání řešeno součinností elektromotoru a tlačné pružiny tak, že elektromotor prostřednictvím šnekové převodovky ovládá předpětí pružiny.

Parametry robotizovaného podvozku

První fází vývoje je tvorba měřítkového modelu, jehož základní parametry jsou uvedeny v tabulce 1. Pokud se model osvědčí, další fází bude tvorba prototypu, jehož rozměry nebudou přímo násobky modelu, ale jednotlivé části budou zvětšeny v požadovaném měřítku, které bude vycházet z finálních rozměrových požadavků. Rozvor a rozchod jsou vzhledem k pohybovým možnostem podvozku značně variabilní, proto jsou v tabulce uvedeny rozměry v mezních polohách.

Parametry modelu	
max. rychlost	8 km/h
pohotovostní hmotnost	20 kg
celková hmotnost	30 kg
rozvor	(100-500)mm
rozchod	(200-600)mm

Parametry konečného provedení budou přibližně odpovídat rozměrům běžně prodávaných vozíků. Celková hmotnost bude cca 200kg, rozvor a rozchod v základní poloze přibližně 1m. Rychlost pohybu bude asi 8km/h.

Zaměření práce

Tato diplomová práce je zaměřena na tvorbu matematického modelu robotizovaného podvozku. Volně navazuje na diplomovou práci Ondřeje Medůny z roku 2002 („Mechanika podvozku se čtyřmi nezávisle zavěšenými koly s tlumiči a pružinami s řízeným předpětím“), který úspěšně implementoval do 2D podoby podvozku regulační soustavu tak, že při pojezdu po zvlněné křivce si užitečné zatížení zachovává stabilizovanou polohu. Ve shodě se zadáním se ale v této práci regulací zabývat nebudeme, protože by to znamenalo řešit úlohu zrychlení s rozsahem výpočtů aktuálně nerealizovatelným.

Cílem práce je vytvoření matematického modelu. Úloha bude řešena v rozsahu úlohy polohy a úlohy rychlosti. Pohyb podvozku bude odvozen od otáčení kol při maximálním možném splnění podmínek valení mezi koly a podložkou.

Matematický model bude mít v budoucnosti následující využití:

- pro nalezení parametrů pohonů, převodů a návrh manévrů,
- bude součástí řídicího systému, kdy bude možné nahradit informace získané z nějakého čidla informacemi získanými z matematického modelu,
- pro vytvoření simulátoru, který bude sloužit k osvojení si ovládání vozidla při každodenním používání.

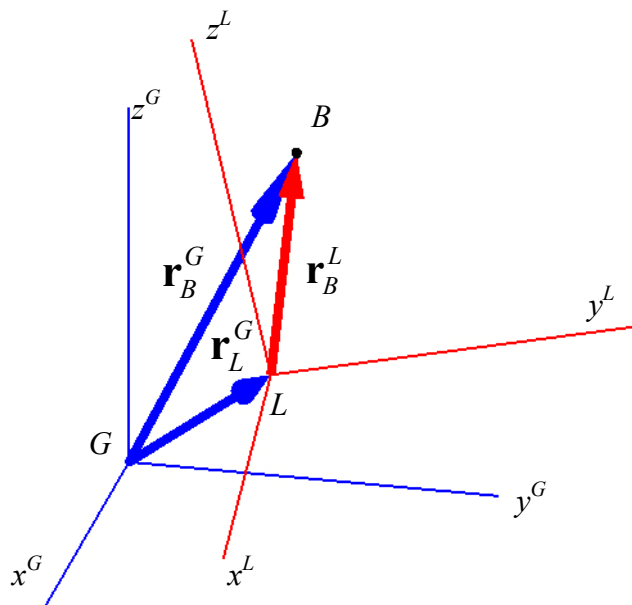
1 Použité matematické postupy

Při řešení zadané úlohy byly použity jisté matematické postupy, které budou přiblíženy v této kapitole. Při tvorbě této kapitoly jsem čerpal z knih uvedených v seznamu použité literatury [1], [2], [3].

1.1 Maticová metoda v kinematice

Jako základní matematický aparát je použit maticový popis matematického modelu. Použity jsou jak matice o rozměru 3×3 a vektory o rozměru 3×1 , tak o rozměru 4×4 a 4×1 , což jsou tzv. rozšířené matice a vektory, které usnadňují popis systému.

1.1.1 Kinematika tělesa ve 3D



Obr.1.1: Souřadnicové systémy

Sledujeme pohyb tělesa, v němž jsme si zvolili lokální souřadnicový systém (L, x^L, y^L, z^L) , v globálním souřadnicovém systému (G, x^G, y^G, z^G) , (Obr.1.1).

Pohyb obecného bodu B tělesa reprezentovaného souřadnicovým systémem (L, x^L, y^L, z^L) je popsán rovnicemi

$$\begin{aligned}x_B^G &= x_B^L \cos \alpha_{x^L} + y_B^L \cos \alpha_{y^L} + z_B^L \cos \alpha_{z^L} + x_L^G, \\y_B^G &= x_B^L \cos \beta_{x^L} + y_B^L \cos \beta_{y^L} + z_B^L \cos \beta_{z^L} + y_L^G, \\z_B^G &= x_B^L \cos \gamma_{x^L} + y_B^L \cos \gamma_{y^L} + z_B^L \cos \gamma_{z^L} + z_L^G,\end{aligned}\tag{1.1}$$

kde

x_B^G, y_B^G, z_B^G jsou souřadnice bodu B v globálním souřadnicovém systému,

x_B^L, y_B^L, z_B^L jsou souřadnice téhož bodu v lokálním souřadnicovém systému, tj. v prostoru tělesa,

$\alpha_{x^L}, \beta_{x^L}, \gamma_{x^L}, (\alpha_{y^L}, \beta_{y^L}, \gamma_{y^L}, \alpha_{z^L}, \beta_{z^L}, \gamma_{z^L})$ jsou směrové úhly, které svírají osy x^G, y^G, z^G s osami x^L, y^L, z^L . Např. β_{x^L} je úhel, který svírá osa x^L a osa y^G , γ_{y^L} je úhel mezi osami y^L a z^G a

x_L^G, y_L^G, z_L^G jsou souřadnice počátku L vyjádřené v globálním souřadnicovém systému.

Uvedené rovnice jsou parametrické rovnice trajektorie bodu B , které se maticově zapíší

$$\begin{bmatrix} x_B^G \\ y_B^G \\ z_B^G \end{bmatrix} = \begin{bmatrix} \cos \alpha_{x^L} & \cos \alpha_{y^L} & \cos \alpha_{z^L} \\ \cos \beta_{x^L} & \cos \beta_{y^L} & \cos \beta_{z^L} \\ \cos \gamma_{x^L} & \cos \gamma_{y^L} & \cos \gamma_{z^L} \end{bmatrix} \begin{bmatrix} x_B^L \\ y_B^L \\ z_B^L \end{bmatrix} + \begin{bmatrix} x_L^G \\ y_L^G \\ z_L^G \end{bmatrix}\tag{1.2}$$

nebo symbolicky

$$\mathbf{r}_B^G = \mathbf{T}^{GL} \mathbf{r}_B^L + \mathbf{r}_L^G,\tag{1.3}$$

kde

$\mathbf{r}_B^G = [x_B^G, y_B^G, z_B^G]^T$ je polohový vektor bodu B v globálním prostoru,

$\mathbf{r}_B^L = [x_B^L, y_B^L, z_B^L]^T$ je polohový vektor téhož bodu v prostoru tělesa,

$\mathbf{r}_L^G = [x_L^G, y_L^G, z_L^G]^T$ je polohový vektor počátku L v globálním prostoru a

$$\mathbf{T}^{GL} = \begin{bmatrix} \cos \alpha_{x^L} & \cos \alpha_{y^L} & \cos \alpha_{z^L} \\ \cos \beta_{x^L} & \cos \beta_{y^L} & \cos \beta_{z^L} \\ \cos \gamma_{x^L} & \cos \gamma_{y^L} & \cos \gamma_{z^L} \end{bmatrix} \quad \text{je transformační}$$

matice z lokálního do globálního souřadnicového systému. Je to matice směrových kosinů, jednotlivé sloupce vyjadřují postupně směrové kosiny os x^L, y^L, z^L . Tato matice je ortogonální, takže platí

$$(\mathbf{T}^{GL})^{-1} = (\mathbf{T}^{GL})^T. \quad (1.4)$$

Rovnice (1.3) jsou maticové rovnice pohybu (trajektorie) bodu B . Obecně jsou $\mathbf{T}^{GL}, \mathbf{r}_B^G, \mathbf{r}_L^G$ funkcemi času a \mathbf{r}_B^L je konstantní vektor, protože bod B se pohybuje spolu s tělesem reprezentovaným lokálním souřadnicovým systémem. Přesněji se proto rovnice (1.3) zapíší jako

$$\mathbf{r}_B^G(t) = \mathbf{T}^{GL}(t) \mathbf{r}_B^L + \mathbf{r}_L^G(t), \quad (1.5)$$

kde

$$\mathbf{r}_B^G(t) = [x_B^G(t), y_B^G(t), z_B^G(t)]^T,$$

$$\mathbf{r}_L^G(t) = [x_L^G(t), y_L^G(t), z_L^G(t)]^T \text{ a}$$

$$\mathbf{T}^{GL}(t) = \begin{bmatrix} \cos \alpha_{x^L}(t) & \cos \alpha_{y^L}(t) & \cos \alpha_{z^L}(t) \\ \cos \beta_{x^L}(t) & \cos \beta_{y^L}(t) & \cos \beta_{z^L}(t) \\ \cos \gamma_{x^L}(t) & \cos \gamma_{y^L}(t) & \cos \gamma_{z^L}(t) \end{bmatrix}.$$

Vztah pro rychlost bodu B dostaneme derivací rovnice (1.5) podle času, s ohledem na to, že $\dot{\mathbf{r}}_B^G = \mathbf{0}$:

$$\mathbf{v}_B^G(t) = \dot{\mathbf{r}}_B^G(t) = \dot{\mathbf{T}}^{GL}(t) \mathbf{r}_B^L + \dot{\mathbf{r}}_L^G(t), \quad (1.6)$$

kde

$\dot{\mathbf{r}}_B^G(t) = [\dot{x}_B^G(t), \dot{y}_B^G(t), \dot{z}_B^G(t)]^T$ je rychlost bodu B vyjádřená v globálním souřadnicovém systému,

$\dot{\mathbf{r}}_L^G(t) = [\dot{x}_L^G(t), \dot{y}_L^G(t), \dot{z}_L^G(t)]^T$ je rychlost bodu L vyjádřená v globálním souřadnicovém systému a

$$\dot{\mathbf{T}}^{GL}(t) = \frac{d}{dt} \begin{bmatrix} \cos \alpha_{x^L}(t) & \cos \alpha_{y^L}(t) & \cos \alpha_{z^L}(t) \\ \cos \beta_{x^L}(t) & \cos \beta_{y^L}(t) & \cos \beta_{z^L}(t) \\ \cos \gamma_{x^L}(t) & \cos \gamma_{y^L}(t) & \cos \gamma_{z^L}(t) \end{bmatrix} \text{ je první derivace transformační matice}$$

podle času.

Pro úplnost uveďme ještě vzorec pro výpočet zrychlení bodu B , který získáme derivací rovnice (1.6) podle času

$$\mathbf{a}_B^G(t) = \dot{\mathbf{v}}_B^G(t) = \ddot{\mathbf{r}}_B^G(t) = \ddot{\mathbf{T}}^{GL}(t) \mathbf{r}_B^L + \dot{\mathbf{T}}^{GL}(t) \dot{\mathbf{r}}_B^L, \quad (1.7)$$

kde

$\ddot{\mathbf{r}}_B^G(t) = [\ddot{x}_B^G(t), \ddot{y}_B^G(t), \ddot{z}_B^G(t)]^T$ je zrychlení bodu B vyjádřené v globálním souřadnicovém systému,

$\ddot{\mathbf{r}}_L^G(t) = [\ddot{x}_L^G(t), \ddot{y}_L^G(t), \ddot{z}_L^G(t)]^T$ je zrychlení počátku L vyjádřené v globálním souřadnicovém systému a

$$\ddot{\mathbf{T}}^{GL}(t) = \frac{d^2}{dt^2} \begin{bmatrix} \cos \alpha_{x^L}(t) & \cos \alpha_{y^L}(t) & \cos \alpha_{z^L}(t) \\ \cos \beta_{x^L}(t) & \cos \beta_{y^L}(t) & \cos \beta_{z^L}(t) \\ \cos \gamma_{x^L}(t) & \cos \gamma_{y^L}(t) & \cos \gamma_{z^L}(t) \end{bmatrix} \text{ je druhá derivace transformační}$$

matice podle času.

1.1.2 Rozšířené matice

Výše uvedená maticová metoda, která využívá matice 3×3 , je vhodná při použití menšího počtu souřadnicových systémů. Nevýhodou je, že jednotlivé transformační matice nelze nahradit jednou, která by reprezentovala pohyb posledního z řetězce souřadnicových systémů vůči globálnímu prostoru.

Tento problém řeší použití tzv. rozšířených matic. Rozšířené matice jsou matice o rozměru 4×4 a obsahují jak směrové kosiny, tak polohový vektor počátku lokálního souřadnicového systému. Systém tvorby rozšířených matic a jejich použití pro systém s více souřadnicovými systémy je obsahem této podkapitoly.

K rovnicím (1.1) připojíme ještě identitu $1 = 1$. Tyto 4 rovnice lze maticově zapsat jako

$$\begin{bmatrix} x_B^G \\ y_B^G \\ z_B^G \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \alpha_{x^L} & \cos \alpha_{y^L} & \cos \alpha_{z^L} & x_L^G \\ \cos \beta_{x^L} & \cos \beta_{y^L} & \cos \beta_{z^L} & y_L^G \\ \cos \gamma_{x^L} & \cos \gamma_{y^L} & \cos \gamma_{z^L} & z_L^G \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_B^L \\ y_B^L \\ z_B^L \\ 1 \end{bmatrix}. \quad (1.8)$$

V rovnici (1.8) se vyskytují jako submatice matice z (1.3), a proto můžeme tuto rovnici přepsat do tvaru

$$\begin{bmatrix} \mathbf{r}_B^G \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{T}^{GL} & \mathbf{r}_L^G \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_B^L \\ 1 \end{bmatrix} \quad (1.9)$$

nebo symbolicky

$$\bar{\mathbf{r}}_B^G = \bar{\mathbf{T}}^{GL} \bar{\mathbf{r}}_B^L, \quad (1.10)$$

kde

$\bar{\mathbf{r}}_B^G = [x_B^G, y_B^G, z_B^G, 1]^T = [\mathbf{r}_B^G, 1]^T$ je rozšířený polohový vektor bodu B v globálním prostoru,

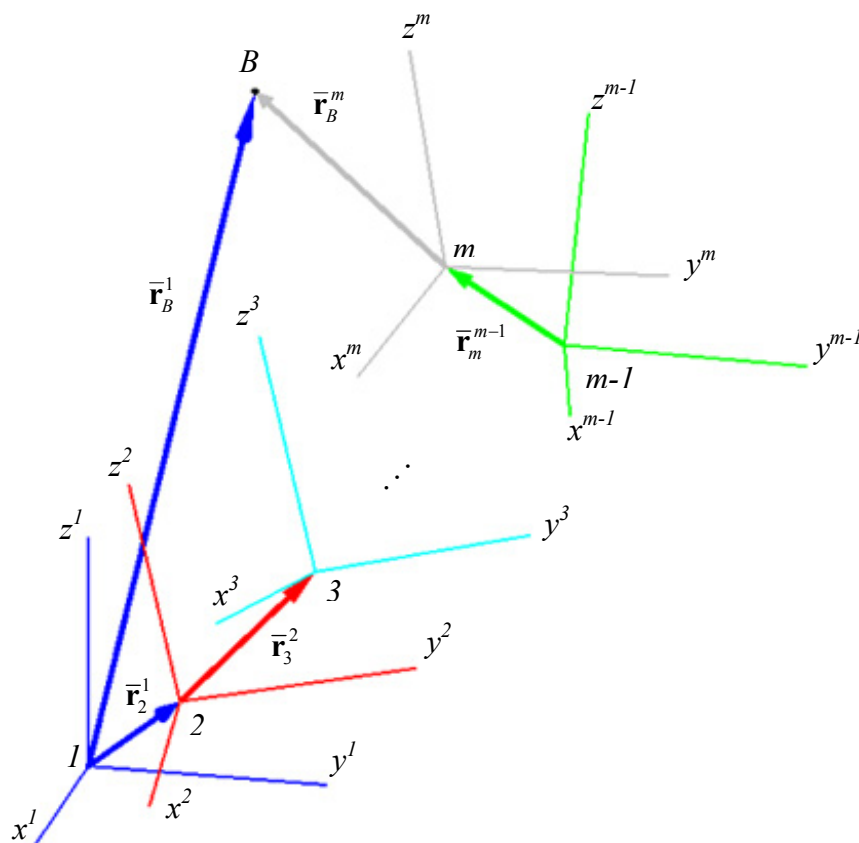
$\bar{\mathbf{r}}_B^L = [x_B^L, y_B^L, z_B^L, 1]^T = [\mathbf{r}_B^L, 1]^T$ je rozšířený polohový vektor bodu B v lokálním prostoru a

$$\bar{\mathbf{T}}^{GL} = \begin{bmatrix} \cos \alpha_{x^L} & \cos \alpha_{y^L} & \cos \alpha_{z^L} & x_L^G \\ \cos \beta_{x^L} & \cos \beta_{y^L} & \cos \beta_{z^L} & y_L^G \\ \cos \gamma_{x^L} & \cos \gamma_{y^L} & \cos \gamma_{z^L} & z_L^G \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{je rozšířená transformační matice.}$$

Pro výpočet rychlosti bodu B použijeme obdobný vztah jako (1.6)

$$\bar{\mathbf{v}}_B^G = \dot{\bar{\mathbf{r}}}_B^G = \dot{\bar{\mathbf{T}}}^{GL} \bar{\mathbf{r}}_B^L \quad (1.11)$$

1.1.2.1 Současné pohyby



Obr.1.2: Řetězec souřadnicových systémů

Pohyb m -tého tělesa soustavy vůči rámu 1 necht' je realizován pomocí současných pohybů popsaných symbolickou rovnicí

$$m:1 = m:(m-1) + (m-1):(m-2) + \dots + 3:2 + 2:1.$$

Vzájemné polohy jednotlivých souřadnicových systémů jsou dány maticemi $\bar{\mathbf{T}}^{m-1,m}, \bar{\mathbf{T}}^{m-2,m-1}, \dots, \bar{\mathbf{T}}^{2,3}, \bar{\mathbf{T}}^{1,2}$ a polohové vektory bodu B jsou vázány podle (1.10) postupně vztahy

$$\begin{aligned} \bar{\mathbf{r}}_B^{m-1} &= \bar{\mathbf{T}}^{m-1,m} \bar{\mathbf{r}}_B^m, \\ \bar{\mathbf{r}}_B^{m-2} &= \bar{\mathbf{T}}^{m-2,m-1} \bar{\mathbf{r}}_B^{m-1}, \\ &\vdots \\ \bar{\mathbf{r}}_B^2 &= \bar{\mathbf{T}}^{2,3} \bar{\mathbf{r}}_B^3, \\ \bar{\mathbf{r}}_B^1 &= \bar{\mathbf{T}}^{1,2} \bar{\mathbf{r}}_B^2. \end{aligned} \tag{1.12}$$

Když z rovnic (1.12) vyloučíme postupně $\bar{\mathbf{r}}_B^2, \bar{\mathbf{r}}_B^3, \dots, \bar{\mathbf{r}}_B^{m-1}$, dostaneme vztah

$$\bar{\mathbf{r}}_B^1 = \bar{\mathbf{T}}^{1,2} \bar{\mathbf{T}}^{2,3} \bar{\mathbf{T}}^{3,4} \dots \bar{\mathbf{T}}^{m-2,m-1} \bar{\mathbf{T}}^{m-1,m} \bar{\mathbf{r}}_B^m. \quad (1.13)$$

Toto je maticová rovnice pohybu bodu B členu m řetězce $1, 2, \dots, m$. Rozepsaná dává tři skalární parametrické rovnice trajektorie bodu B . Čtvrtá rovnice je identita $1 = 1$.

Transformační matice

$$\bar{\mathbf{T}}^{1,m} = \bar{\mathbf{T}}^{1,2} \bar{\mathbf{T}}^{2,3} \bar{\mathbf{T}}^{3,4} \dots \bar{\mathbf{T}}^{m-2,m-1} \bar{\mathbf{T}}^{m-1,m} \quad (1.14)$$

je transformační matice pohybu členu m vzhledem k základnímu rámu 1. Pomocí ní můžeme zapsat vztah (1.13) jako

$$\bar{\mathbf{r}}_B^1 = \bar{\mathbf{T}}^{1,m} \bar{\mathbf{r}}_B^m. \quad (1.15)$$

1.2 Určení normály k ploše

Určení normály k ploše je důležitou součástí této diplomové práce. Pomocí normály k ploše je možné určit polohu středu kola o daném poloměru při jeho odvalování po pojezdové ploše jen ze znalosti bodu dotyku.

Normálu hledáme jako jednotkový vektor ve tvaru vektorové funkce $\mathbf{n}(x, y)$ v bodě o souřadnicích $[x, y, f(x, y)]$. Jednotlivé složky n_x , n_y a n_z vektoru \mathbf{n} jsou v tom případě směrové kosiny, pro které platí

$$\sqrt{n_x^2 + n_y^2 + n_z^2} = 1. \quad (1.16)$$

Pro potřeby výpočtů souvisejících s matematickým modelem vyjádříme pojezdovou plochu explicitně jako $z = f(x, y)$. V tomto případě jsou směrové kosiny normály definovány jako

$$\begin{aligned}
 n_x &= \frac{-p(x, y)}{\sqrt{p(x, y)^2 + q(x, y)^2 + 1}}, \\
 n_y &= \frac{-q(x, y)}{\sqrt{p(x, y)^2 + q(x, y)^2 + 1}}, \\
 n_z &= \frac{1}{\sqrt{p(x, y)^2 + q(x, y)^2 + 1}},
 \end{aligned} \tag{1.17}$$

kde

$$p(x, y) = \frac{\partial f(x, y)}{\partial x}, q(x, y) = \frac{\partial f(x, y)}{\partial y}.$$

1.3 Určení vzdálenosti bodu od přímky

Určení vzdálenosti bodu od přímky bude využito při stanovení poloměru, na kterém dochází k odvalování kulového kola po podložce. Poloměr, na kterém se kolo odvaluje, se mění v závislosti na nastavených hodnotách parametrů podvozku.

Mějme bod o souřadnicích $[x_0, y_0, z_0]$. Hledáme jeho vzdálenost d od přímky tvořené bodem o souřadnicích $[x_1, y_1, z_1]$ a směrovým vektorem $\mathbf{a} = [a_1, a_2, a_3]$. Potom je hledaná vzdálenost

$$d = \frac{|\mathbf{u} \times \mathbf{a}|}{|\mathbf{a}|}, \tag{1.18}$$

kde

$$\mathbf{u} = [x_0 - x_1, y_0 - y_1, z_0 - z_1].$$

1.4 Numerické řešení soustav diferenciálních rovnic

Pro numerické řešení musíme soustavu převést do tzv. kanonického tvaru. Kanonický tvar soustavy diferenciálních rovnic je dán zápisem

$$\begin{aligned} \frac{d^1 y}{d x} &= {}^1 f(x, {}^1 y, \dots, {}^n y), \\ &\dots\dots\dots \\ \frac{d^ n y}{d x} &= {}^n f(x, {}^1 y, \dots, {}^n y), \end{aligned} \quad (1.19)$$

nebo stručněji

$$\frac{d^ j y}{d x} = {}^j f(x, {}^1 y, \dots, {}^n y), \quad j = 1, 2, \dots, n. \quad (1.20)$$

Vlastní numerické řešení soustavy diferenciálních rovnic je možné provést více způsoby. Vzhledem k očekávané složitosti v budoucnu řešených diferenciálních rovnic byla pro výpočet zvolena metoda Runge - Kuttova.

1.4.1 Runge - Kuttova metoda 4. řádu

Po převedení soustavy do kanonického tvaru s použitím počátečních podmínek, můžeme s pomocí následujících vzorců určit funkční hodnoty hledaných neznámých.

$${}^j Y_1 = {}^j Y_0 + \frac{h}{2} ({}^j k_1 + 2 {}^j k_2 + 2 {}^j k_3 + {}^j k_4), \quad j = 1, 2, \dots, n, \quad (1.21)$$

kde

$$\begin{aligned} {}^j k_1 &= {}^j f(x_0, {}^1 Y_0, \dots, {}^n Y_0), \\ {}^j k_2 &= {}^j f\left(x_0 + \frac{h}{2}, {}^1 Y_0 + \frac{h}{2} {}^1 k_1, \dots, {}^n Y_0 + \frac{h}{2} {}^n k_1\right), \\ {}^j k_3 &= {}^j f\left(x_0 + \frac{h}{2}, {}^1 Y_0 + \frac{h}{2} {}^1 k_2, \dots, {}^n Y_0 + \frac{h}{2} {}^n k_2\right), \\ {}^j k_4 &= {}^j f(x_0 + h, {}^1 Y_0 + h {}^1 k_3, \dots, {}^n Y_0 + h {}^n k_3), \end{aligned} \quad (1.22)$$

${}^j Y_0$ je hodnota j -té neznámé na začátku kroku,

jY_1 je hodnota j -té neznámé na konci kroku,

n je počet neznámých,

h je krok nezávisle proměnné.

Nevýhodou této metody je velká časová náročnost výpočtu u složitějších soustav rovnic.

1.5 Software Maple

Matematický model byl vytvořen v softwaru Maple. Maple je počítačové prostředí, které bylo vyvinuto na univerzitě Waterloo v Kanadě, pro zjednodušení a zrychlení výpočtů v matematice. Na rozdíl od klasických programů pro numerické výpočty modeluje matematické operace se symbolickými výrazy. Maple umožňuje provádět jak symbolické a numerické výpočty, tak vytvářet grafy funkcí, programovat vlastní funkce či procedury, ukládat data v několika formátech (např. *LaTeX*, *HTML*, *RTF*, *MATHML*, ...) a dokonce i provádět export do jiných programovacích jazyků.

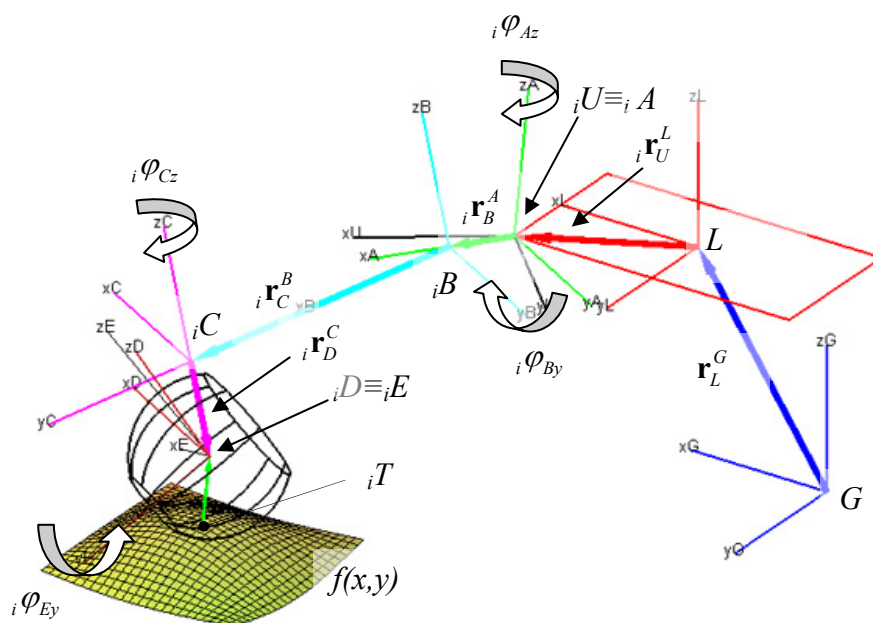
Funkce implementované v Maplu pokrývají širokou oblast matematiky od základů lineární algebry, diferenciálního a integrálního počtu, přes diferenciální rovnice, geometrii až k logice.

2 Souřadnicové systémy

Celý podvozek se skládá z více součástí a tyto součásti mají mezi sebou určité vazby. Jednotlivé součásti jsou v matematickém modelu reprezentovány různými souřadnicovými systémy. Vazby mezi jednotlivými součástmi vyjadřují transformační matice.

Pro vyjádření všech možných pohybů všech částí podvozku bylo použito celkem 26 souřadnicových systémů. Z toho 5 souřadnicových systémů bylo použito na každou nohu, čtyři reprezentují místo připojení nohy k podvozku, jeden reprezentuje podvozek robotu a jeden náleží globálnímu prostoru.

Na Obr.2.1 je znázorněn řetězec souřadnicových systémů, které přísluší i -té noze, podvozku robotu a globálnímu prostoru. Jsou zde také vyznačeny polohové vektory počátků souřadnicových systémů a jednotlivé stupně volnosti, které budou popsány v následujícím textu.



Obr.2.1: Souřadnicové systémy a stupně volnosti

2.1 Globální souřadnicový systém GSS

Globální souřadnicový systém (dále jen GSS) je základním souřadnicovým systémem celého matematického modelu. Je spojen s podložkou, po které robot pojíždí a má v čase konstantní polohu. Vůči tomuto systému se realizují všechny pohyby ostatních částí podvozku robotu.

2.2 Souřadnicový systém (L, x^L, y^L, z^L)

Tento souřadnicový systém představuje podvozek robotu, ke kterému jsou připojeny 4 výkyvné nohy. Počátek tohoto systému je v GSS určen vektorem (viz Obr.2.1)

$$\mathbf{r}_L^G = \begin{bmatrix} x_L^G \\ y_L^G \\ z_L^G \end{bmatrix}. \quad (2.1)$$

Podvozek vykonává obecný prostorový pohyb se šesti stupni volnosti, které se vztahují ke třem souřadnicím x_L^G, y_L^G, z_L^G počátku L a ke třem sférickým úhlům $\varphi_x, \varphi_y, \varphi_z$. Používají se různé systémy těchto úhlů:

- Systém Eulerův (tzv. “3-1-3”) - jedná se o pootočení o úhel ψ - *precese* okolo osy z , o úhel ϑ - *nutace* okolo nové osy x a o úhel φ - *vlastní rotace* okolo pootočené osy z . Tento systém má nevýhodu v tom, že když je úhel ϑ roven nule, tak nelze rozlišit úhly ψ a φ , jedná se o tzv. singularitu.
- Systém Cardanův (“1-2-3“, BODY FIXED) - pootočení se měří okolo osy x o úhel Φ_x , potom okolo osy y o úhel Φ_y a následovně okolo osy z o úhel Φ_z
- Systém RPY (“1-2-3“, FIXED FRAME) - tento systém je podobný předchozímu případu s tím rozdílem, že úhly $\varphi_x, \varphi_y, \varphi_z$ se vztahují k rotacím okolo os rovnoběžných s osami GSS.

Pro stanovení polohy souřadnicového systému (L, x^L, y^L, z^L) byl zvolen systém sférických úhlů RPY, jak je obvyklé u vozidel, letadel a plavidel.

2.2.1 Systém sférických úhlů RPY

Jak už bylo uvedeno, systém sférických úhlů používá k popsání relativního sférického pohybu tří úhlů $\varphi_x, \varphi_y, \varphi_z$. Zkratka RPY (Roll - Pitch - Yaw), v překladu klonění - klopení - zatáčení, vychází z názvosloví jednotlivých rotací kolem os souřadnicového systému letadel.

Transformační matice relativního sférického pohybu je v tomto případě

$$\mathbf{T} = \mathbf{T}(\varphi_x, \varphi_y, \varphi_z). \quad (2.2)$$

Tato matice vznikne součinem matic (v obráceném pořadí než u systému Cardanových úhlů), které představují jednotlivé rotace okolo souřadnicových os.

První maticí je rotace o úhel φ_x okolo osy x . Tato matice je definovaná jako

$$\mathbf{T}^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi_x & -\sin \varphi_x \\ 0 & \sin \varphi_x & \cos \varphi_x \end{bmatrix}. \quad (2.3)$$

Druhou maticí je rotace o úhel φ_y okolo osy y . Tato matice je definovaná následovně

$$\mathbf{T}^2 = \begin{bmatrix} \cos \varphi_y & 0 & \sin \varphi_y \\ 0 & 1 & 0 \\ -\sin \varphi_y & 0 & \cos \varphi_y \end{bmatrix}. \quad (2.4)$$

Poslední maticí je rotace o úhel φ_z okolo osy z definovaná

$$\mathbf{T}^3 = \begin{bmatrix} \cos \varphi_z & -\sin \varphi_z & 0 \\ \sin \varphi_z & \cos \varphi_z & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.5)$$

Výsledná transformační matice je pak

$$\mathbf{T} = \mathbf{T}^3 \mathbf{T}^2 \mathbf{T}^1, \quad (2.6)$$

neboli pro transformaci souřadnic ze systému (L, x^L, y^L, z^L) do GSS máme matici

$$\mathbf{T}^{GL} = \begin{bmatrix} \cos \varphi_z \cos \varphi_y & -\sin \varphi_z \cos \varphi_x + \cos \varphi_z \sin \varphi_y \sin \varphi_x & \sin \varphi_z \sin \varphi_x + \cos \varphi_z \sin \varphi_y \cos \varphi_x \\ \sin \varphi_z \cos \varphi_y & \cos \varphi_z \cos \varphi_x + \sin \varphi_z \sin \varphi_y \sin \varphi_x & -\cos \varphi_z \sin \varphi_x + \sin \varphi_z \sin \varphi_y \cos \varphi_x \\ -\sin \varphi_y & \cos \varphi_y \sin \varphi_x & \cos \varphi_y \cos \varphi_x \end{bmatrix}. \quad (2.7)$$

Z (2.1) a (2.7) podle (1.9) složíme rozšířenou transformační matici ze systému (L, x^L, y^L, z^L) do GSS

$$\bar{\mathbf{T}}^{GL} = \begin{bmatrix} \cos\varphi_z \cos\varphi_y & -\sin\varphi_z \cos\varphi_x + \cos\varphi_z \sin\varphi_y \sin\varphi_x & \sin\varphi_z \sin\varphi_x + \cos\varphi_z \sin\varphi_y \cos\varphi_x & {}_i x_L^G \\ \sin\varphi_z \cos\varphi_y & \cos\varphi_z \cos\varphi_x + \sin\varphi_z \sin\varphi_y \sin\varphi_x & -\cos\varphi_z \sin\varphi_x + \sin\varphi_z \sin\varphi_y \cos\varphi_x & {}_i y_L^G \\ -\sin\varphi_y & \cos\varphi_y \sin\varphi_x & \cos\varphi_y \cos\varphi_x & {}_i z_L^G \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.8)$$

2.3 Souřadnicové systémy ${}_i(U, x^U, y^U, z^U)$, $i=1,2,\dots,4$

Souřadnicový systém ${}_i(U, x^U, y^U, z^U)$ reprezentuje místo připojení i -té výkyvné nohy k podvozku robotu a je znázorněn na Obr.2.1. Tento souřadnicový systém s počátkem v bodě ${}_iU$ je vůči souřadnicovému systému (L, x^L, y^L, z^L) pootočen o úhel ${}_i\varphi_{Uz}$ okolo osy z^L a následně pak okolo nově vzniklé osy y^L pootočen o úhel ${}_i\varphi_{Uy}$. Úhly ${}_i\varphi_{Uz}$ a ${}_i\varphi_{Uy}$ jsou konstrukční úhly.

Polohový vektor počátku souřadnicového systému ${}_i(U, x^U, y^U, z^U)$ je v souřadnicovém systému (L, x^L, y^L, z^L) vyjádřen jako vektor

$${}_i \mathbf{r}_U^L = \begin{bmatrix} {}_i x_U^L \\ {}_i y_U^L \\ {}_i z_U^L \end{bmatrix}. \quad (2.9)$$

Transformační matice ze systému ${}_i(U, x^U, y^U, z^U)$ do (L, x^L, y^L, z^L) vznikne jako součin dvou transformačních matic, které reprezentují uvedené rotace. Výsledná rozšířená transformační matice bude ve tvaru

$${}_i \bar{\mathbf{T}}^{LU} = \begin{bmatrix} \cos({}_i\varphi_{Uz})\cos({}_i\varphi_{Uy}) & -\sin({}_i\varphi_{Uz}) & \cos({}_i\varphi_{Uz})\sin({}_i\varphi_{Uy}) & {}_i x_U^L \\ \sin({}_i\varphi_{Uz})\cos({}_i\varphi_{Uy}) & \cos({}_i\varphi_{Uz}) & \sin({}_i\varphi_{Uz})\sin({}_i\varphi_{Uy}) & {}_i y_U^L \\ -\sin({}_i\varphi_{Uy}) & 0 & \cos({}_i\varphi_{Uy}) & {}_i z_U^L \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.10)$$

Vynásobením matice ${}_i \bar{\mathbf{T}}^{LU}$ zleva maticí $\bar{\mathbf{T}}^{GL}$ podle (1.14) dostáváme transformační matici ze systému ${}_i(U, x^U, y^U, z^U)$ do GSS

$${}_i \bar{\mathbf{T}}^{GU} = \bar{\mathbf{T}}^{GL} {}_i \bar{\mathbf{T}}^{LU}. \quad (2.11)$$

2.4 Souřadnicové systémy $_i(A, x^A, y^A, z^A)$, $i=1,2,\dots,4$

Počátek $_iA$ souřadnicového systému $_i(A, x^A, y^A, z^A)$ i -té podvozkové nohy je vůči systému $_i(U, x^U, y^U, z^U)$ vyjádřen polohovým vektorem

$$_i\mathbf{r}_A^U = \begin{bmatrix} _i x_A^U \\ _i y_A^U \\ _i z_A^U \end{bmatrix} \quad (2.12)$$

a je pootočen o úhel $_i\varphi_{Az}$ okolo osy u^Z , Obr.2.1. Úhly $_i\varphi_{Az}$, $i=1,2,\dots,4$ se nazývají rejdy a jejich změny jsou ovládány samostatnými elektromotory. Změna těchto úhlů bude potřeba např. při překonávání překážky nebo při změně rozchodu či rozvoru podvozku.

Rozšířená transformační matice bude pro transformaci tohoto systému do $_i(U, x^U, y^U, z^U)$ v následujícím tvaru

$$_i\overline{\mathbf{T}}^{UA} = \begin{bmatrix} \cos(_i\varphi_{Az}) & -\sin(_i\varphi_{Az}) & 0 & _i x_A^U \\ \sin(_i\varphi_{Az}) & \cos(_i\varphi_{Az}) & 0 & _i y_A^U \\ 0 & 0 & 1 & _i z_A^U \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.13)$$

Transformace do GSS bude podle (1.14)

$$_i\overline{\mathbf{T}}^{GA} = \overline{\mathbf{T}}^{GL} {}_i\overline{\mathbf{T}}^{LU} {}_i\overline{\mathbf{T}}^{UA} \quad (2.14)$$

nebo s využitím (2.11)

$$_i\overline{\mathbf{T}}^{GA} = {}_i\overline{\mathbf{T}}^{GU} {}_i\overline{\mathbf{T}}^{UA}. \quad (2.15)$$

2.5 Souřadnicové systémy $_i(B, x^B, y^B, z^B)$, $i=1,2,\dots,4$

Na Obr.2.1 je také vyznačen souřadnicový systém $_i(B, x^B, y^B, z^B)$ s počátkem určeným polohovým vektorem

$$_i\mathbf{r}_B^A = \begin{bmatrix} _i x_B^A \\ _i y_B^A \\ _i z_B^A \end{bmatrix}. \quad (2.16)$$

Tento souřadnicový systém je vůči souřadnicovému systému $_i(A, x^A, y^A, z^A)$ pootočen okolo osy y^A o úhel ${}_i\varphi_{By}$ a přísluší i -té noze robotizovaného podvozku. Změnou těchto úhlů budou řídicím systémem vyrovnávány nerovnosti terénu tak, aby byl podvozek stabilizován ve vodorovné poloze. Změnou úhlů ${}_i\varphi_{By}, i=1,2,\dots,4$ bude také možné měnit světlou výšku podvozku dle požadavků situace. Ovládání úhlů ${}_i\varphi_{By}, i=1,2,\dots,4$ je unikátně řešeno prostřednictvím elektromotorů a pružin, kde elektromotorem ovládáme předpětí pružiny. Tím je zajištěno, že i při nastaveném předpětí pružin jsou dovoleny určité změny úhlů ${}_i\varphi_{By}, i=1,2,\dots,4$, které mohou eliminovat nerovnosti terénu tak malé, že na ně nezareaguje regulační soustava.

Rozšířená transformační matice ze systému $_i(B, x^B, y^B, z^B)$ do $_i(A, x^A, y^A, z^A)$ s využitím (2.16) je

$${}_i\overline{\mathbf{T}}^{AB} = \begin{bmatrix} \cos({}_i\varphi_{By}) & 0 & \sin({}_i\varphi_{By}) & {}_ix_B^A \\ 0 & 1 & 0 & {}_iy_B^A \\ -\sin({}_i\varphi_{By}) & 0 & \cos({}_i\varphi_{By}) & {}_iz_B^A \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.17)$$

Pro výpočet transformační matice z $_i(B, x^B, y^B, z^B)$ do GSS použijeme opět vztah (1.14). Výsledná transformační matice bude pak mít tvar

$${}_i\overline{\mathbf{T}}^{GB} = \overline{\mathbf{T}}^{GL} {}_i\overline{\mathbf{T}}^{LU} {}_i\overline{\mathbf{T}}^{UA} {}_i\overline{\mathbf{T}}^{AB} \quad (2.18)$$

nebo s využitím (2.15)

$${}_i\overline{\mathbf{T}}^{GB} = {}_i\overline{\mathbf{T}}^{GA} {}_i\overline{\mathbf{T}}^{AB}. \quad (2.19)$$

2.6 Souřadnicové systémy $_i(C, x^C, y^C, z^C), i=1,2,\dots,4$

Dalším souřadnicovým systémem je souřadnicový systém s počátkem v bodě ${}_iC$ i -té nohy, jehož polohový vektor je v $_i(B, x^B, y^B, z^B)$ vyjádřen jako

$${}_i\mathbf{r}_C^B = \begin{bmatrix} {}_ix_C^B \\ {}_iy_C^B \\ {}_iz_C^B \end{bmatrix}. \quad (2.20)$$

Tento souřadnicový systém má vzhledem k systému ${}_i(B, x^B, y^B, z^B)$ jeden stupeň volnosti a to rotaci (pivotaci) kolem osy z^B o úhel ${}_i\varphi_{Cz}$. Rotace okolo os z^B o úhly ${}_i\varphi_{Cz}$, $i=1,2,\dots,4$ jsou zajištěny samostatnými elektromotory a jsou potřeba při změně směru jízdy. Úhel pivotace je uveden na Obr.2.1.

Rozšířená transformační matice z ${}_i(C, x^C, y^C, z^C)$ do ${}_i(B, x^B, y^B, z^B)$ je s využitím (2.5) a (2.20) podle (1.9) dána vztahem

$${}_i\bar{\mathbf{T}}^{BC} = \begin{bmatrix} \cos({}_i\varphi_{Cz}) & \sin({}_i\varphi_{Cz}) & 0 & {}_i x_C^B \\ -\sin({}_i\varphi_{Cz}) & \cos({}_i\varphi_{Cz}) & 0 & {}_i y_C^B \\ 0 & 0 & 1 & {}_i z_C^B \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.21)$$

Podle vztahu (1.14) určíme rozšířenou transformační matici, která charakterizuje pohyb tohoto souřadnicového systému v GSS

$${}_i\bar{\mathbf{T}}^{GC} = \bar{\mathbf{T}}^{GL} {}_i\bar{\mathbf{T}}^{LU} {}_i\bar{\mathbf{T}}^{UA} {}_i\bar{\mathbf{T}}^{AB} {}_i\bar{\mathbf{T}}^{BC} \quad (2.22)$$

nebo

$${}_i\bar{\mathbf{T}}^{GC} = {}_i\bar{\mathbf{T}}^{GB} {}_i\bar{\mathbf{T}}^{BC}. \quad (2.23)$$

2.7 Souřadnicové systémy ${}_i(D, x^D, y^D, z^D)$, $i=1,2,\dots,4$

Souřadnicový systém s počátkem v bodě ${}_iD$, což je střed i -tého kola, je posunut o ${}_i z_D^C$ vzhledem k ${}_i(C, x^C, y^C, z^C)$ a je pootočen o úhel ${}_i\varphi_{Dx}$ okolo osy x^C souřadnicového systému ${}_i(C, x^C, y^C, z^C)$. Souřadnice ${}_i x_D^C$, ${}_i y_D^C$ jsou rovny nule proto, aby poloha bodu dotyku kola (tvořeného kulovou plochou) s pojezdovou plochou nezáležela na pootočení souřadnicového systému ${}_i(C, x^C, y^C, z^C)$ o úhel ${}_i\varphi_{Cz}$. Pootočení těchto souřadnicových systémů o úhly ${}_i\varphi_{Dx}$, $i=1,2,\dots,4$ jsou provedena jen z konstrukčních důvodů a podobně jako úhly ${}_i\varphi_{Uz}$ a ${}_i\varphi_{Uy}$ nezajišťují žádný z požadovaných pohybů robotu.

Polohový vektor počátku ${}_iD$ je ve tvaru

$${}_i\mathbf{r}_D^C = \begin{bmatrix} 0 \\ 0 \\ {}_i z_D^C \end{bmatrix}. \quad (2.24)$$

Rozšířená transformační matice ze systému ${}_i(D, x^D, y^D, z^D)$ do ${}_i(C, x^C, y^C, z^C)$ bude mít v případě posunutí o ${}_i\mathbf{r}_D^C$ a rotace o ${}_i\varphi_{Dx}$ tvar

$${}_i\overline{\mathbf{T}}^{CD} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos({}_i\varphi_{Dx}) & -\sin({}_i\varphi_{Dx}) & 0 \\ 0 & \sin({}_i\varphi_{Dx}) & \cos({}_i\varphi_{Dx}) & {}_i z_D^C \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.25)$$

a transformace do GSS bude analogicky

$${}_i\overline{\mathbf{T}}^{GD} = \overline{\mathbf{T}}^{GL} {}_i\overline{\mathbf{T}}^{LU} {}_i\overline{\mathbf{T}}^{UA} {}_i\overline{\mathbf{T}}^{AB} {}_i\overline{\mathbf{T}}^{BC} {}_i\overline{\mathbf{T}}^{CD} \quad (2.26)$$

nebo

$${}_i\overline{\mathbf{T}}^{GD} = {}_i\overline{\mathbf{T}}^{GC} {}_i\overline{\mathbf{T}}^{CD}. \quad (2.27)$$

2.8 Souřadnicové systémy ${}_i(E, x^E, y^E, z^E)$, $i=1,2,\dots,4$

Bod ${}_iE$, který představuje počátek souřadnicového systému ${}_i(E, x^E, y^E, z^E)$, je totožný s bodem ${}_iD$. Tento souřadnicový systém je pevně spojen s i -tým kolem a otáčí se vůči ${}_i(D, x^D, y^D, z^D)$ kolem společné osy $y^D \equiv y^E$ o úhel ${}_i\varphi_{Ey}$. Změna úhlů ${}_i\varphi_{Ey}$, $i=1,2,\dots,4$ je ovládána elektromotory pojezdu vozidla.

Rozšířená transformační matice z ${}_i(E, x^E, y^E, z^E)$ do ${}_i(D, x^D, y^D, z^D)$ je

$${}_i\overline{\mathbf{T}}^{DE} = \begin{bmatrix} \cos({}_i\varphi_{Ey}) & 0 & \sin({}_i\varphi_{Ey}) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin({}_i\varphi_{Ey}) & 0 & \cos({}_i\varphi_{Ey}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.28)$$

a transformace do GSS je zajištěna maticí

$${}_i\overline{\mathbf{T}}^{GE} = \overline{\mathbf{T}}^{GL} {}_i\overline{\mathbf{T}}^{LU} {}_i\overline{\mathbf{T}}^{UA} {}_i\overline{\mathbf{T}}^{AB} {}_i\overline{\mathbf{T}}^{BC} {}_i\overline{\mathbf{T}}^{CD} {}_i\overline{\mathbf{T}}^{DE} \quad (2.29)$$

nebo s využitím (2.27)

$${}_i\bar{\mathbf{T}}^{GE} = {}_i\bar{\mathbf{T}}^{GD} {}_i\bar{\mathbf{T}}^{DE}. \quad (2.30)$$

2.9 Body dotyku i -tého kola s pojezdovou plochou $f(x,y)$

Na Obr.2.1 je bod dotyku i -tého kola s pojezdovou plochou reprezentován bodem ${}_iT$, který je v GSS určen polohovým vektorem

$${}_i\bar{\mathbf{r}}_T^G = \begin{bmatrix} {}_i x_T^G \\ {}_i y_T^G \\ f({}_i x_T^G, {}_i y_T^G) \\ 1 \end{bmatrix}. \quad (2.31)$$

Složky ${}_i x_T^G$ a ${}_i y_T^G$ vektoru ${}_i\bar{\mathbf{r}}_T^G$ budou předmětem výpočtu, viz kapitola 3.

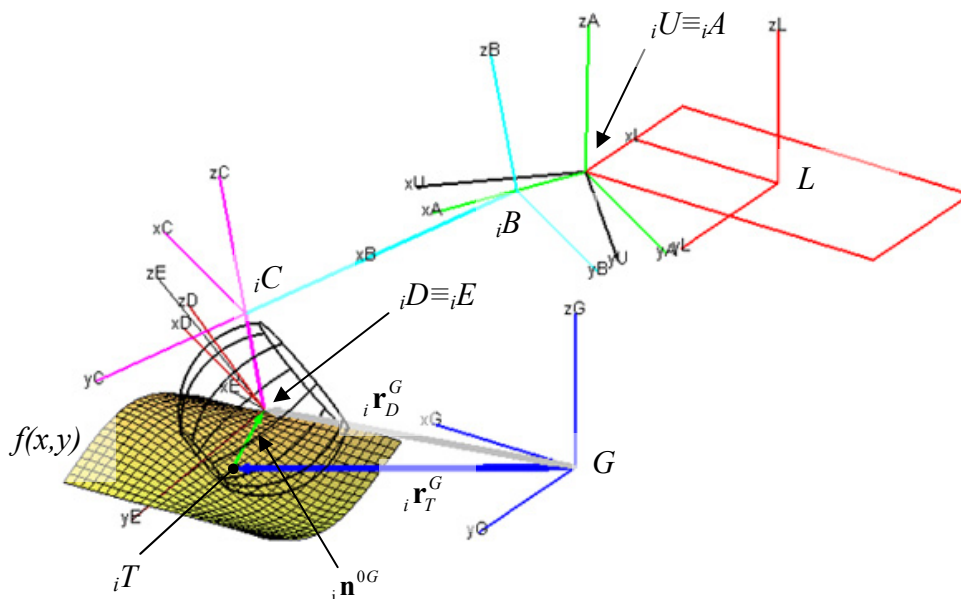
3 Vlastní matematický model

V této kapitole budou uvedeny základní vztahy pro popis matematického modelu. Cílem bude, aby byl pohyb celého podvozku odvozen od požadované rychlosti pohybu a zadaného poloměru zatačky, což jsou dva parametry, které budou ovládány uživatelem.

Většina polohových vektorů, určujících počátky souřadnicových systémů na Obr.2.1, je konstantní, protože jsou dány geometrií konstrukce podvozkové nohy. Jedná se o vektory ${}_i\mathbf{r}_U^L$, ${}_i\mathbf{r}_B^A$, ${}_i\mathbf{r}_C^B$, ${}_i\mathbf{r}_D^C$, $i=1,2,\dots,4$. Výjimku tvoří vektor \mathbf{r}_L^G , který definuje polohu bodu L v GSS, tedy vlastně polohu vozidla na pojezdové ploše. Tento vektor má tři nekonstantní nenulové složky. Z nich x -ová a y -ová složka, tedy x_L^G a y_L^G , budou předmětem výpočtu a z -ová složka, která definuje světlou výšku podvozku, bude v budoucnu řízena systémem automatické stabilizace. Stabilizační systém ovšem nyní není součástí modelu, takže zde bude zadána jako funkce času $z_L^G = z_L^G(t)$. Jako funkce času musí být zadány i veličiny, které ve výpočtu figurují jako vstupní parametry. Jsou to polohové parametry příslušné stupňům volnosti ovládaným samostatnými elektromotory: ${}_i\varphi_{Az} = {}_i\varphi_{Az}(t)$, ${}_i\varphi_{Cz} = {}_i\varphi_{Cz}(t)$, ${}_i\varphi_{Ey} = {}_i\varphi_{Ey}(t)$, $i=1,2,\dots,4$. Výjimku tvoří úhly ${}_i\varphi_{By}$, $i=1,2,\dots,4$, které budou, podobně jako x_L^G , y_L^G , ${}_i x_T^G$ a ${}_i y_T^G$, předmětem výpočtu. Hodnoty úhlů ${}_i\varphi_{By}$, $i=1,2,\dots,4$, budou počítány za předpokladu, že sférické úhly φ_x , φ_y jsou rovny nule. Ve skutečnosti bude nulování úhlů φ_x , φ_y zajištěno opět systémem automatické stabilizace, pro který budou φ_x , φ_y regulované veličiny a úhly ${}_i\varphi_{By}$, $i=1,2,\dots,4$ veličiny akční. Zbývá tedy určit úhel φ_z , což je poslední parametr, který získáme výpočtem.

Proveďme nyní „inventuru“ neznámých. Po zadání všech známých délkových a úhlových rozměrů zbývá určit φ_z , x_L^G , y_L^G , pro polohu rámu vozidla a ${}_i x_T^G$, ${}_i y_T^G$, ${}_i\varphi_{By}$ pro nohy $i=1,2,\dots,4$, tedy celkem 15 hodnot pro každý časový okamžik. Způsob jejich výpočtu bude uveden v následujícím textu.

3.1 Úloha polohy



Obr.3.1: Vektory týkající se úlohy polohy

Cílem bude určit úhly ${}_i\varphi_{By}$ a vektory ${}_i\mathbf{r}_T^G$, při zadaných ostatních úhlových a délkových rozměrech tak, aby se každé kolo $i = 1, 2, \dots, 4$ stýkalo s podpůrnou plochou $f(x, y)$ vždy jen v jednom bodě a to v bodě ${}_iT$. Okamžité hodnoty veličin ${}_i\varphi_{Az}, i = 1, 2, \dots, 4, z_L^G, \varphi_z, x_L^G, y_L^G$, budeme přitom považovat za zadané a hodnoty ${}_i\varphi_{Ey}, {}_i\varphi_{Cz}, i = 1, 2, \dots, 4$, nemají na polohu bodu dotyku i -tého kola žádný vliv, jak vyplývá z uspořádání souřadnicových systémů. Proto budeme psát transformační rovnice pro body ${}_iD$ a to přes souřadnicové systémy $(L, x^L, y^L, z^L), {}_i(U, x^U, y^U, z^U), {}_i(A, x^A, y^A, z^A), {}_i(B, x^B, y^B, z^B), {}_i(C, x^C, y^C, z^C), {}_i(D, x^D, y^D, z^D)$. Následně budeme polohu bodů ${}_iD$ v GSS definovat pomocí normály k ploše v bodě ${}_iT$ a poloměru kola, Obr.3.1.

Polohy bodů ${}_iD, i = 1, 2, \dots, 4$, vyjádřené v GSS, reprezentované vektory ${}_i\bar{\mathbf{r}}_D^G, i = 1, 2, \dots, 4$, určíme jako

$${}_i\bar{\mathbf{r}}_D^G = {}_i\bar{\mathbf{T}}^{GD} {}_i\bar{\mathbf{r}}_D^D, \quad i = 1, 2, \dots, 4, \quad (3.1)$$

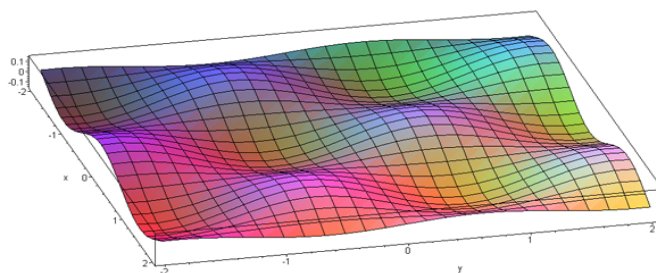
kde

$${}_i\bar{\mathbf{r}}_D^D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \text{ vyjadřují polohové vektory bodů } {}_iD \text{ v systémech } {}_i(D, x^D, y^D, z^D), \quad i = 1, 2, \dots, 4.$$

Pro výpočet normály k ploše, kterou budeme potřebovat pro sestavení druhé maticové rovnice určující polohu bodů ${}_iD$, $i = 1, 2, \dots, 4$, využijeme vztahu (1.17). Pojezdovou plochu $f(x, y)$ budeme uvažovat zcela obecně. Pro testovací účely byla uvažována jako

$$f(x, y) = F_0 \cos \frac{x}{X_0} \sin \frac{y}{Y_0}. \quad (3.2)$$

Konkrétní tvar plochy viz Obr.3.2.



Obr.3.2: Tvar testovací pojezdové plochy pro $F_0 = 0,15$, $X_0 = Y_0 = 0,4$.

Jednotkové normálové vektory s počátkem v bodech ${}_iT$, $i = 1, 2, \dots, 4$ s polohovými vektory ${}_i\mathbf{r}_T^G$, $i = 1, 2, \dots, 4$, budou

$${}_i\mathbf{n}^{0G} = \begin{bmatrix} -\frac{\left.\frac{\partial f(x,y)}{\partial x}\right|_{x={}_ix_T^G, y={}_iy_T^G}}{\sqrt{\left(\left.\frac{\partial f(x,y)}{\partial x}\right|_{x={}_ix_T^G, y={}_iy_T^G}\right)^2 + \left(\left.\frac{\partial f(x,y)}{\partial y}\right|_{x={}_ix_T^G, y={}_iy_T^G}\right)^2} + 1} \\ -\frac{\left.\frac{\partial f(x,y)}{\partial y}\right|_{x={}_ix_T^G, y={}_iy_T^G}}{\sqrt{\left(\left.\frac{\partial f(x,y)}{\partial x}\right|_{x={}_ix_T^G, y={}_iy_T^G}\right)^2 + \left(\left.\frac{\partial f(x,y)}{\partial y}\right|_{x={}_ix_T^G, y={}_iy_T^G}\right)^2} + 1} \\ 1 \end{bmatrix}, \quad i = 1, 2, \dots, 4. \quad (3.3)$$

Protože jsou transformační rovnice (3.1) vyjádřeny pomocí rozšířených matic a vektorů, musí být vektory ${}_i\mathbf{n}^{0G}$, $i = 1, 2, \dots, 4$ uvedeny také v rozšířeném tvaru, tedy

$${}_i\bar{\mathbf{n}}^{0G} = \begin{bmatrix} {}_i\mathbf{n}^{0G} \\ 1 \end{bmatrix}, \quad i = 1, 2, \dots, 4. \quad (3.4)$$

Poloměr kola označíme r a s použitím (3.4) vyjádříme polohy bodů ${}_iD$ jako

$${}_i\bar{\mathbf{r}}_D^G = {}_i\bar{\mathbf{r}}_T^G + r {}_i\bar{\mathbf{n}}^{0G}, \quad i = 1, 2, \dots, 4. \quad (3.5)$$

Rovnice (3.1) a (3.5) vyjadřují polohu stejného bodu, takže můžeme přejít k rovnosti pravých stran

$${}_i\bar{\mathbf{T}}^{GD} {}_i\bar{\mathbf{r}}_D^D = {}_i\bar{\mathbf{r}}_T^G + r {}_i\bar{\mathbf{n}}^{0G}, \quad i = 1, 2, \dots, 4. \quad (3.6)$$

Tím získáváme soustavu 4×3 rovnic pro výpočet dvanácti neznámých ${}_i\varphi_{By}$, ${}_ix_T^G$, ${}_iy_T^G$, $i = 1, 2, \dots, 4$. (Třetí složku vektorů ${}_i\bar{\mathbf{r}}_T^G$, $i = 1, 2, \dots, 4$, určíme z rovnice plochy, tedy ${}_iz_T^G = f({}_ix_T^G, {}_iy_T^G)$, $i = 1, 2, \dots, 4$.) V případě, že uvažujeme jen tyto neznámé, rozpadne se soustava rovnic tak, že pro každé i máme soustavu tří rovnic pro tři neznámé.

Jak bylo uvedeno v počátku kapitoly, můžeme takto uvažovat jen v případě, kdy ${}_i\varphi_{Az}$, ${}_iz_L^G$, φ_z , ${}_ix_L^G$, ${}_iy_L^G$, $i = 1, 2, \dots, 4$, jsou známé hodnoty. Ve skutečnosti jsou však

${}_i\varphi_{Az} = {}_i\varphi_{Az}(t)$, $i = 1, 2, \dots, 4$ a $z_L^G = z_L^G(t)$ dané funkce času a hodnoty φ_z , x_L^G , y_L^G , které určují okamžitou polohu a pozici vozidla, se mění v důsledku pohonu kol. To má dva důsledky. Zaprvé jsou neznámé φ_z , x_L^G , y_L^G obsaženy ve všech rovnicích soustavy (3.6), takže k rozpadu na 3 x 4 rovnice nedochází. A za druhé tři rovnice chybí.

3.2 Úloha rychlosti

Protože kola se po jezdové ploše odvalují, budou tři chybějící rovnice v soustavě (3.6) doplněny v podobě podmínek valení. Podmínky valení budou vyjádřeny jako nulové rychlosti určitých bodů v určitých směrech, tj. bude se jednat o diferenciální rovnice prvního řádu. Výsledná soustava 15 rovnic tak bude algebro - diferenciální. Proto musíme i soustavu (3.6) derivovat podle času, abychom získali soustavu 15 diferenciálních rovnic, kterou už budeme schopni vyřešit.

3.2.1 Derivace soustavy (3.6) podle času

Soustava rovnic (3.6) je nyní ve tvaru

$${}_i\bar{\mathbf{T}}^{GD}(t, x_L^G, y_L^G, \varphi_z, \varphi_{By}) {}_i\bar{\mathbf{r}}_D^D = {}_i\bar{\mathbf{r}}_T^G(x_T^G, y_T^G) + r_i \bar{\mathbf{n}}^{0G}(x_T^G, y_T^G), \quad i = 1, 2, \dots, 4. \quad (3.7)$$

Derivací podle času přejde do tvaru

$${}_i\dot{\bar{\mathbf{T}}}^{GD} {}_i\bar{\mathbf{r}}_D^D = {}_i\dot{\bar{\mathbf{r}}}_T^G + r_i \dot{\bar{\mathbf{n}}}^{0G}, \quad i = 1, 2, \dots, 4, \quad (3.8)$$

kde

$${}_i\dot{\bar{\mathbf{T}}}^{GD} = \frac{\partial {}_i\bar{\mathbf{T}}^{GD}}{\partial t} + \frac{\partial {}_i\bar{\mathbf{T}}^{GD}}{\partial x_L^G} \dot{x}_L^G + \frac{\partial {}_i\bar{\mathbf{T}}^{GD}}{\partial y_L^G} \dot{y}_L^G + \frac{\partial {}_i\bar{\mathbf{T}}^{GD}}{\partial \varphi_z} \dot{\varphi}_z + \frac{\partial {}_i\bar{\mathbf{T}}^{GD}}{\partial \varphi_{By}} \dot{\varphi}_{By},$$

$${}_i\dot{\bar{\mathbf{r}}}_T^G = \frac{\partial {}_i\bar{\mathbf{r}}_T^G}{\partial x_T^G} \dot{x}_T^G + \frac{\partial {}_i\bar{\mathbf{r}}_T^G}{\partial y_T^G} \dot{y}_T^G,$$

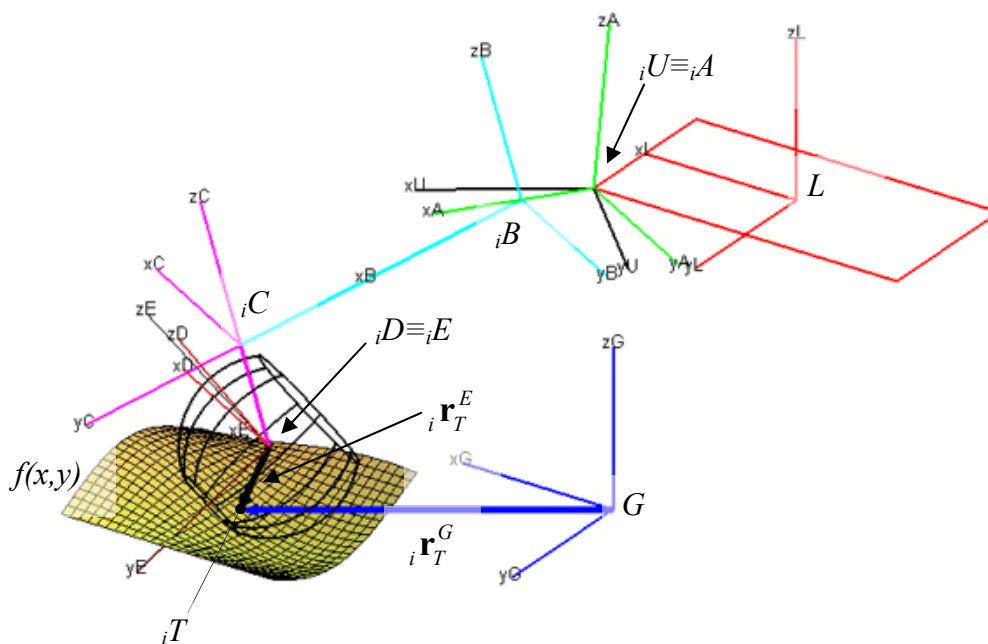
$$\dot{\bar{\mathbf{n}}}^{0G} = \frac{\partial \bar{\mathbf{n}}^{0G}}{\partial x_T^G} \dot{x}_T^G + \frac{\partial \bar{\mathbf{n}}^{0G}}{\partial y_T^G} \dot{y}_T^G,$$

takže jde o soustavu 12 diferenciálních rovnic 1. řádu pro 15 neznámých funkcí $x_L^G(t), y_L^G(t), \varphi_z(t), {}_i x_T^G(t), {}_i y_T^G(t), {}_i \varphi_{By}(t), i = 1, 2, \dots, 4$.

3.2.2 Podmínky valení

Podmínky valení budeme definovat pro bod dotyku i -tého kola s pojezdovou plochou $f(x, y)$. Při dokonalém valení tělesa kulového tvaru je rychlost dotykového bodu rovna nule, neboli bod dotyku je okamžitým pólem, kolem kterého se kulová plocha odvaluje.

3.2.2.1 Rychlost dotykového bodu považovaného za bod i -tého kola



Obr.3.2: Vektory týkající se úlohy rychlosti

Prvním krokem je vyjádření polohových vektorů ${}_i \vec{r}_T^E$, $i = 1, 2, \dots, 4$, což jsou polohové vektory bodů dotyku ${}_i T$, $i = 1, 2, \dots, 4$, vyjádřené v rotujících souřadnicových systémech ${}_i(E, x^E, y^E, z^E)$, $i = 1, 2, \dots, 4$. K tomu použijeme polohové vektory bodů ${}_i T$, $i = 1, 2, \dots, 4$, vyjádřené v GSS a transformační matice ze souřadnicových systémů ${}_i(E, x^E, y^E, z^E)$, $i = 1, 2, \dots, 4$, do GSS. Platí, že

$${}_i\bar{\mathbf{r}}_T^G = {}_i\bar{\mathbf{T}}^{GE} {}_i\bar{\mathbf{r}}_T^E, \quad i = 1, 2, \dots, 4. \quad (3.9)$$

Z rovnice (3.9) vyjádříme vektory ${}_i\bar{\mathbf{r}}_T^E$, $i = 1, 2, \dots, 4$, vynásobením zleva maticí $({}_i\bar{\mathbf{T}}^{GE})^{-1}$ s využitím vztahu $({}_i\bar{\mathbf{T}}^{GE})^{-1} = {}_i\bar{\mathbf{T}}^{EG}$

$${}_i\bar{\mathbf{r}}_T^E = {}_i\bar{\mathbf{T}}^{EG} {}_i\bar{\mathbf{r}}_T^G, \quad i = 1, 2, \dots, 4, \quad (3.10)$$

kde

${}_i\bar{\mathbf{T}}^{EG}$ je transformační matice z GSS do systému ${}_i(E, x^E, y^E, z^E)$ pevně spojeného s i -tým kolem.

Vektory rychlostí dotykových bodů ${}_iT$, kde si bod ${}_iT$ představíme jako součást i -tého kola, získáme derivací rovnice (3.10) podle času za předpokladu, že budeme uvažovat vektory ${}_i\bar{\mathbf{r}}_T^E$, $i = 1, 2, \dots, 4$, neproměnné v čase, tj. jako by měly v každém okamžiku vůči bodům ${}_iE$ konstantní polohu, tedy $\dot{{}_i\bar{\mathbf{r}}_T^E} = \mathbf{0}$. Označíme-li je jako $({}_i\bar{\mathbf{v}}_T^G)^E$, potom

$$({}_i\bar{\mathbf{v}}_T^G)^E = \dot{{}_i\bar{\mathbf{T}}^{GE}} {}_i\bar{\mathbf{r}}_T^E, \quad i = 1, 2, \dots, 4, \quad (3.11)$$

kde

$$\dot{{}_i\bar{\mathbf{T}}^{GE}} = \frac{\partial {}_i\bar{\mathbf{T}}^{GE}}{\partial t} + \frac{\partial {}_i\bar{\mathbf{T}}^{GE}}{\partial x_L^G} \dot{x}_L^G + \frac{\partial {}_i\bar{\mathbf{T}}^{GE}}{\partial y_L^G} \dot{y}_L^G + \frac{\partial {}_i\bar{\mathbf{T}}^{GE}}{\partial \varphi_z} \dot{\varphi}_z + \frac{\partial {}_i\bar{\mathbf{T}}^{GE}}{\partial \varphi_{By}} \dot{\varphi}_{By}.$$

Po dosazení z (3.10) za ${}_i\bar{\mathbf{r}}_T^E$ do (3.11) dostáváme konečné vztahy pro rychlosti $({}_i\bar{\mathbf{v}}_T^G)^E$, $i = 1, 2, \dots, 4$,

$$({}_i\bar{\mathbf{v}}_T^G)^E = \dot{{}_i\bar{\mathbf{T}}^{GE}} {}_i\bar{\mathbf{T}}^{EG} {}_i\bar{\mathbf{r}}_T^G, \quad i = 1, 2, \dots, 4. \quad (3.12)$$

3.2.2.2 Přiřazení podmínek valení

Vektory rychlostí $({}_i\bar{\mathbf{v}}_T^G)^E$, $i = 1, 2, \dots, 4$, mají v GSS obecně tři nenulové složky. Kdybychom si tyto vektory vyjádřili v takových souřadnicových systémech, které by měly dvě souřadnicové osy v tečné rovině k ploše $f(x, y)$ v bodě ${}_iT$ a třetí osu kolmou na předchozí dvě, tj. ve směru normály k tečné rovině, potom by vektory $({}_i\bar{\mathbf{v}}_T^G)^E$,

$i = 1, 2, \dots, 4$, měly pro každé i pouze dvě nenulové složky. To je způsobeno tím, že jsou kola svázána s plochou $f(x, y)$ podmínkou dotyku v jednom bodě.

Uplatnění podmínky valení na všechna čtyři kola by proto představovalo dalších osm rovnic. V soustavě (3.8) ale chybí pouze rovnice tři, z čehož plyne, že v kinematickém modelu není možné požadovat splnění podmínek valení na všech kolech současně.

Z tohoto důvodu použijeme jen tři podmínky valení, které uplatníme na dvou kolech. U prvního kola budeme uvažovat dokonalé valení, které odebírá dva stupně volnosti (podmínky valení). Druhé kolo bude podmínku valení splňovat pouze v jednom směru a v druhém bude mít dovolený prokluz. Prokluz bude umožněn ve směru osy rotace kola.

3.2.2.3 Podmínka valení u dokonale se valícího kola

Podmínku budeme definovat pro první kolo ($i = 1$). Pro toto kolo musí platit

$$\left({}_1\mathbf{v}_T^G\right)^E = \mathbf{0}. \quad (3.13)$$

Zde využijeme faktu, že složka rychlosti $\left({}_1\mathbf{v}_T^G\right)^E$ ve směru normály k ploše $f(x, y)$ je rovna nule, a protože musí platit podmínka (3.13), potom i průmět rychlosti $\left({}_1\mathbf{v}_T^G\right)^E$ do jakýchkoli dvou směrů, s výjimkou směru normály, určených dvěma lineárně nezávislými vektory musí být roven nule.

Pro vyjádření těchto podmínek valení musíme nejprve transformovat vektor $\left({}_1\mathbf{v}_T^G\right)^E$ z GSS do souřadnicového systému ${}_1(D, x^D, y^D, z^D)$. Pro transformaci použijeme matici ${}_1\mathbf{T}^{DG}$, což je inverzní matice k již zmíněné matici ${}_1\mathbf{T}^{GD}$, tedy

$$\left({}_1\mathbf{v}_T^D\right)^E = {}_1\mathbf{T}^{DG}\left({}_1\mathbf{v}_T^G\right)^E. \quad (3.14)$$

Osy každého souřadnicového systému, jak vyplývá z definice, musí být tvořeny lineárně nezávislými vektory. Proto můžeme podmínky valení uvažovat jako nulové

složky vektoru rychlosti $\left({}_1\mathbf{v}_T^D\right)^E$ ve směru os x^D a y^D . Tím vzniknou rovnice

$$\left({}_1\mathbf{v}_T^D\right)_x^E = 0, \quad (3.15)$$

$$\left({}_1\mathbf{v}_T^D\right)_y^E = 0, \quad (3.16)$$

které doplní soustavu (3.8).

Složka vektoru ve směru osy z^D nebyla použita, protože se tato osa může svou orientací blížit ke směru normály.

3.2.2.4 Podmínka valení u kola, které má definovanou podmínku valení v daném směru

Podmínku pro druhé kolo ($i = 2$) budeme definovat podobným způsobem jako v předchozím případě pouze s rozdílem, že za nulovou budeme považovat jen složku ve směru osy x^D souřadnicového systému ${}_2(D, x^D, y^D, z^D)$. Opět si vyjádříme vektor $\left({}_2\mathbf{v}_T^G\right)^E$ v souřadnicovém systému ${}_2(D, x^D, y^D, z^D)$

$$\left({}_2\mathbf{v}_T^D\right)^E = {}_2\mathbf{T}^{DG}\left({}_2\mathbf{v}_T^G\right)^E. \quad (3.17)$$

Podmínka, že

$$\left({}_2\mathbf{v}_T^D\right)_x^E = 0, \quad (3.18)$$

je poslední chybějící rovnicí v soustavě (3.8).

Složka vektoru $\left({}_2\mathbf{v}_T^D\right)^E$ ve směru osy x^D byla použita proto, že pokud bude tato složka rovna nule, potom se může vektor $\left({}_2\mathbf{v}_T^G\right)^E$ nacházet jen v rovině tvořené osami y^D a z^D . Vektor $\left({}_2\mathbf{v}_T^G\right)^E$ se musí také nacházet v tečné rovině k ploše $f(x,y)$ v bodě dotyku. Obě tyto podmínky nám zajistí, že kolo má dovolený prokluz v požadovaném směru.

3.3 Vlastní řešení soustavy

V této fázi je nutné řešit soustavu devíti diferenciálních rovnic oproti původním patnácti. To je způsobeno tím, že jsou podmínky valení určeny jen pro první a druhé kolo. To znamená, že bude pohyb podvozku odvozen od otáčení těchto dvou kol.

Jedná se o maticové rovnice (3.8) pro $i = 1, 2$ a rovnice (3.15), (3.16), (3.18).

$$\begin{aligned} {}_1\dot{\mathbf{T}}^{GD} {}_1\bar{\mathbf{r}}_D^D &= {}_1\dot{\mathbf{r}}_T^G + r_1 \dot{\mathbf{n}}^{0G}, \\ {}_2\dot{\mathbf{T}}^{GD} {}_2\bar{\mathbf{r}}_D^D &= {}_2\dot{\mathbf{r}}_T^G + r_2 \dot{\mathbf{n}}^{0G}, \\ ({}_1\mathbf{v}_T^D)_x^E &= 0, \\ ({}_1\mathbf{v}_T^D)_y^E &= 0, \\ ({}_2\mathbf{v}_T^D)_x^E &= 0. \end{aligned} \quad (3.19)$$

Po zadání všech konstrukčních rozměrů a všech parametrů, jež jsou uvažovány jako funkce času, se v rovnicích (3.23) vyskytují následující neznámé a jejich derivace

$${}_1\phi_{By}, {}_1\dot{\phi}_{By}, {}_1x_T^G, {}_1\dot{x}_T^G, {}_1y_T^G, {}_1\dot{y}_T^G, {}_2\phi_{By}, {}_2\dot{\phi}_{By}, {}_2x_T^G, {}_2\dot{x}_T^G, {}_2y_T^G, {}_2\dot{y}_T^G, x_L^G, \dot{x}_L^G, y_L^G, \dot{y}_L^G, \phi_z, \dot{\phi}_z.$$

Pro řešení soustavy (3.19) byla zvolena metoda Runge - Kuttova podle vzoru (1.21). Jak bylo uvedeno v kapitole 1.4, musíme soustavu převést do kanonického tvaru daného předpisem (1.19). Zde bude využito toho, že soustava (3.19) je lineární vůči derivacím ${}_1\dot{\phi}_{By}, {}_1\dot{x}_T^G, {}_1\dot{y}_T^G, {}_2\dot{\phi}_{By}, {}_2\dot{x}_T^G, {}_2\dot{y}_T^G, \dot{x}_L^G, \dot{y}_L^G, \dot{\phi}_z$. Proto můžeme soustavu přepsat do maticového tvaru

$$\mathbf{A}\mathbf{q} = \mathbf{a}, \quad (3.20)$$

kde

$$\begin{aligned} \mathbf{A} &= \mathbf{A}({}_1\phi_{By}, {}_1x_T^G, {}_1y_T^G, {}_2\phi_{By}, {}_2x_T^G, {}_2y_T^G, x_L^G, y_L^G, \phi_z), \\ \mathbf{q} &= [{}_1\dot{\phi}_{By}, {}_1\dot{x}_T^G, {}_1\dot{y}_T^G, {}_2\dot{\phi}_{By}, {}_2\dot{x}_T^G, {}_2\dot{y}_T^G, \dot{x}_L^G, \dot{y}_L^G, \dot{\phi}_z]^T \mathbf{a} \\ \mathbf{a} &= \mathbf{a}({}_1\phi_{By}, {}_1x_T^G, {}_1y_T^G, {}_2\phi_{By}, {}_2x_T^G, {}_2y_T^G, x_L^G, y_L^G, \phi_z). \end{aligned}$$

Soustavu převedeme do kanonického tvaru tak, že z maticové rovnice (3.20) vyjádříme \mathbf{q} . Jednotlivé složky \mathbf{q} budou potom představovat funkce ${}^j f$ použité v (1.22)

$$\mathbf{q} = \mathbf{A}^{-1}\mathbf{a}. \quad (3.21)$$

3.3.1 Počáteční podmínky pro soustavu (3.23)

Nyní už jen zbývá určit počáteční podmínky, abychom mohli soustavu (3.19) vyřešit. K jejich určení použijeme maticovou rovnici (3.6), opět pro $i = 1, 2$. Protože tyto dvě maticové rovnice obsahují všech devět neznámých, budeme muset tři z nich zvolit. Zvolíme výchozí polohu a pozici vozidla, tj. x_L^G, y_L^G, φ_z , a dopočítáme.

$${}_1\varphi_{By}, {}_1x_T^G, {}_1y_T^G, {}_2\varphi_{By}, {}_2x_T^G, {}_2y_T^G.$$

Aby bylo možné vykreslit pohybující se podvozek, je potřeba dopočítat polohové a úhlové hodnoty, týkající se třetího a čtvrtého kola. Potřebné hodnoty spočítáme z rovnic (3.6) pro $i = 3, 4$. Neznámými pro nás budou ${}_3\varphi_{By}, {}_3x_T^G, {}_3y_T^G, {}_4\varphi_{By}, {}_4x_T^G, {}_4y_T^G$. Jestliže nás budou zajímat i derivace těchto hodnot podle času, tedy ${}_3\dot{\varphi}_{By}, {}_3\dot{x}_T^G, {}_3\dot{y}_T^G, {}_4\dot{\varphi}_{By}, {}_4\dot{x}_T^G, {}_4\dot{y}_T^G$, tak použijeme rovnici (3.8) pro $i = 3, 4$.

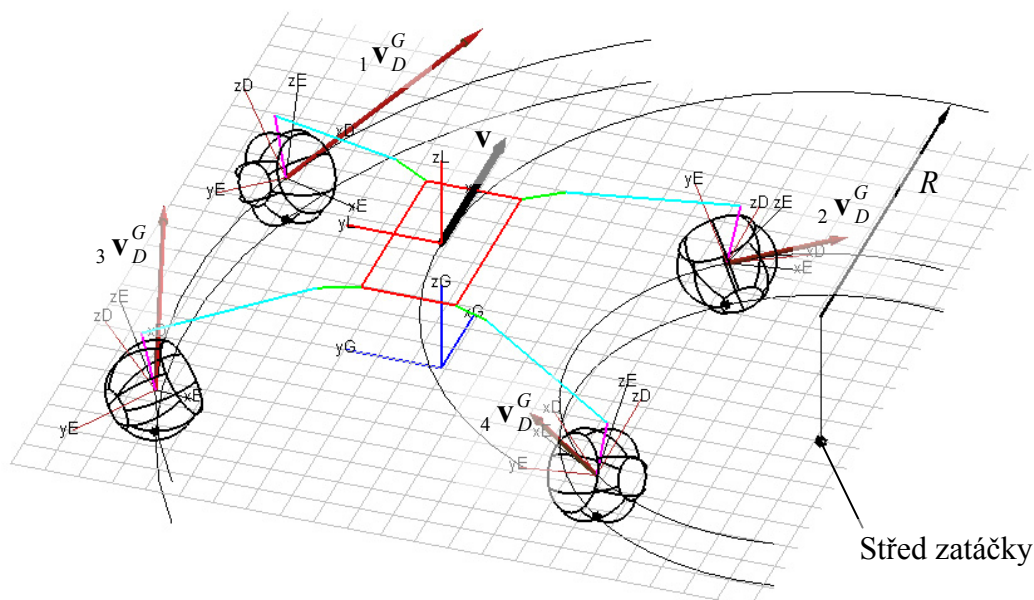
Výsledkem tedy je soustava devíti diferenciálních rovnic, kterou numericky řešíme metodou Runge - Kutta. Zajímají-li nás ještě hodnoty derivací veličin vypočtených touto metodou, stačí pouze dosadit do jednotlivých funkcí ${}^j f$, vytvořených převodem soustavy (3.19) do kanonického tvaru.

4 Základní manévry

V úvodu bylo zmíněno, jaké základní pohyby je robotizovaný podvozek díky své konfiguraci schopen provádět. V této kapitole bude popsáno, co se při takovýchto manévrech s podvozkem děje a když to bude možné, budou zde uvedeny rovnice, podle kterých by se mělo dosáhnout požadovaného pohybu. O přesné formulaci algoritmů bude rozhodnuto později po úspěšném odladění a odzkoušení na funkčním modelu, nebo až se matematický model doplní o úlohu zrychlení a bude možné takové manévry provádět s ohledem na všechny vlivy, jako je např. setrvačná síla, poloha těžiště, atd.

4.1 Jízda do zatáčky

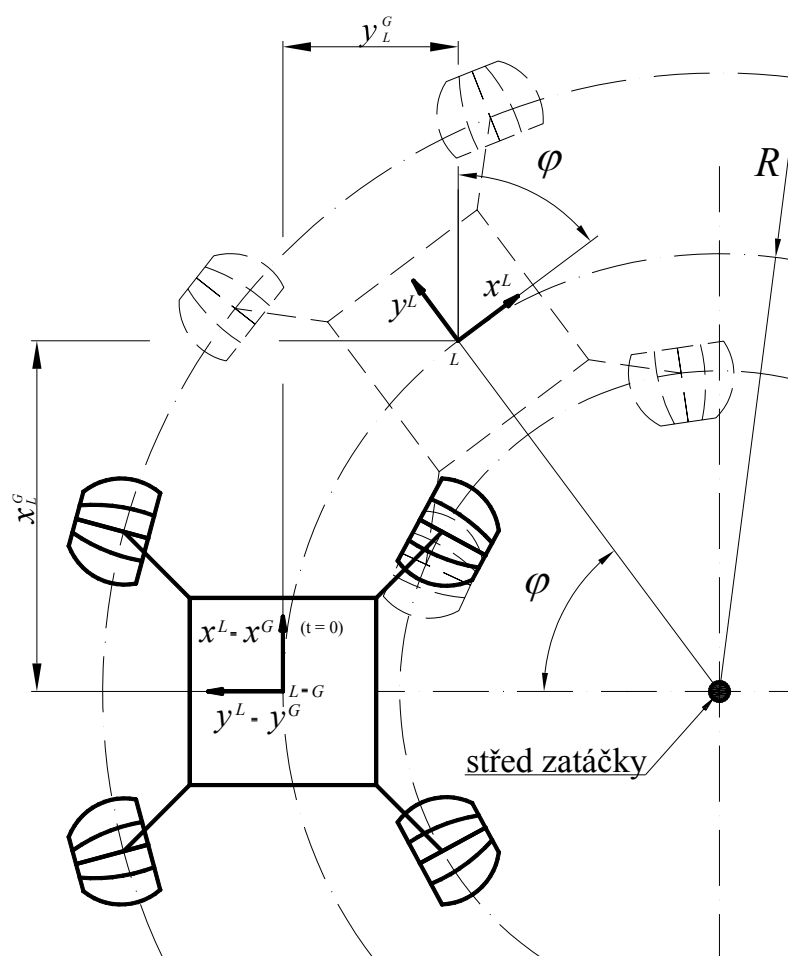
Jízda do zatáčky bude u podvozku nejvíce užívaným manévrem. Cílem této podkapitoly bude odvodit natočení kol (pivotaci) a úhlové rychlosti otáčení kol, aby se podvozek pohyboval po kruhové zatáčce o daném poloměru R danou rychlostí v . V tomto případě budeme uvažovat rovnou pojezdovou plochu.



Obr.4.1: Požadované rychlosti bodů D_i

Na Obr.4.1 je vyznačena požadovaná rychlost v průjezdu zatáčkou a také je zde vyznačen požadovaný střed zatáčky o poloměru R . Dále jsou zde vyznačeny požadované rychlosti bodů D , které se mění v závislosti na hodnotách v , R a na nastavených parametrech podvozku.

Jak už bylo uvedeno, budeme uvažovat, že při průjezdu robotizovaného podvozku zatáčkou o poloměru R se bude robot pohybovat po kruhové dráze. Jeho polohu určíme úhlem φ . Z Obr.4.2 je zřejmé, že po ujetí dráhy příslušné nějakému úhlu φ dojde také v rovině x^L, y^L (okolo osy z^L) k pootočení podvozku o tentýž úhel.



Obr.4.2: Průjezd zatáčkou

Nejprve musíme určit změnu úhlu $\varphi = \varphi(t)$. Vztah pro $\varphi(t)$ budeme hledat ve tvaru $\varphi(t) = -\omega t$. Znaménko minus zohledňuje to, že při odvození je brán jako kladný úhel φ úhel narůstající proti směru hodinových ručiček. Uvažujme, že požadovaná rychlost v se

také rovná obvodové rychlosti průjezdu zatáčkou. S použitím vztahů pro pohyb po kružnici určíme úhlovou rychlost tohoto pohybu

$$\omega = \frac{v}{R}. \quad (4.1)$$

Polohu lokálního počátku L vzhledem k GSS určíme pomocí parametrické rovnice kružnice

$$\begin{aligned} x_L^G(t) &= R \sin \varphi(t), \\ y_L^G(t) &= R(\cos \varphi(t) - 1). \end{aligned} \quad (4.2)$$

S využitím (4.1) a (4.2) složíme pomocnou transformační matici ze systému (L, x^L, y^L, z^L) do GSS a označíme ji $\bar{\mathbf{T}}_p^{GL}$

$$\bar{\mathbf{T}}_p^{GL} = \begin{bmatrix} \cos \varphi(t) & -\sin \varphi(t) & 0 & x_L^G(t) \\ \sin \varphi(t) & \cos \varphi(t) & 0 & y_L^G(t) \\ 0 & 0 & 1 & z_L^G(t) \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.3)$$

Pomocí matice $\bar{\mathbf{T}}_p^{GL}$ vyjádříme nepatrně změněný řetězec souřadnicových systémů GSS až ${}_i(D, x^D, y^D, z^D)$

$${}_i\bar{\mathbf{T}}_p^{GD} = \bar{\mathbf{T}}_p^{GL} \bar{\mathbf{T}}_i^{LU} \bar{\mathbf{T}}_i^{UA} \bar{\mathbf{T}}_i^{AB} \bar{\mathbf{T}}_i^{BC} \bar{\mathbf{T}}_i^{CD}, \quad i = 1, 2, \dots, 4. \quad (4.4)$$

Nyní ze soustavy (3.6), kde nahradíme matici ${}_i\bar{\mathbf{T}}^{GD}$ maticí ${}_i\bar{\mathbf{T}}_p^{GD}$, pro daný čas vypočítáme hodnoty ${}_i\varphi_{By}, {}_i x_T^G, {}_i y_T^G, i = 1, 2, \dots, 4$.

Polohový vektor určující polohu bodů ${}_iD$ v průběhu průjezdu zatáčkou vyjádříme podobně jako v (3.1)

$${}_i\bar{\mathbf{r}}_D^G = {}_i\bar{\mathbf{T}}_p^{GD} {}_i\bar{\mathbf{r}}_D^D, \quad i = 1, 2, \dots, 4. \quad (4.5)$$

Všechna kola musí být natočena tak, aby směřovala ve směrech rychlostí bodů ${}_iD$. Vektory těchto rychlostí zjistíme derivací vztahu (4.5) podle času

$${}_i\bar{\mathbf{v}}_D^G = {}_i\dot{\bar{\mathbf{r}}}_D^G = {}_i\dot{\bar{\mathbf{T}}}_p^{GD} {}_i\bar{\mathbf{r}}_D^D, \quad i = 1, 2, \dots, 4. \quad (4.6)$$

Dále musíme sestavit rovnice pro výpočet úhlů ${}_i\varphi_{Cz}$, $i=1,2,\dots,4$. Vyjádříme jednotkové vektory ve směru os y^D souřadnicových systémů ${}_i(D, x^D, y^D, z^D)$, $i=1,2,\dots,4$,

$${}_i\bar{\mathbf{n}}_{Dy}^G = {}_i\bar{\mathbf{T}}_p^{GD} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} - {}_i\bar{\mathbf{T}}_p^{GD} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad i=1,2,\dots,4. \quad (4.7)$$

Rovnice pro výpočet úhlů ${}_i\varphi_{Cz}$, $i=1,2,\dots,4$, vyjádříme jako skalární součiny vektorů rychlostí a jednotkových vektorů ve směrech os y^D z (4.7) položených rovno nule, protože vektory musí svírat úhel $\frac{\pi}{2}$.

$${}_i\mathbf{v}_D^G \cdot {}_i\mathbf{n}_{Dy}^G = 0, \quad i=1,2,\dots,4. \quad (4.8)$$

Z analytického řešení rovnic (4.8) pro ${}_i\varphi_{Cz}$, $i=1,2,\dots,4$, které je ve tvaru $\arctan(p(R))$, vyplývá, že rovnice mají více řešení. Je tedy nutné je řešit numericky v daných mezích, abychom zajistili požadovaný směr natočení kola.

Následujícím krokem je výpočet poloměru otáčení kola. Zde využijeme vztahu (1.18). Při výpočtu tohoto poloměru můžeme využít analytického řešení rovnic (4.8), protože funkce \arctan má periodu π a výsledek rovnic (4.8) změněný o π vede ke stejným poloměrům otáčení.

Poloměry odvalování kol odpovídají vzdálenostem bodů ${}_iT$, $i=1,2,\dots,4$, od přímek tvořených body ${}_iD$, $i=1,2,\dots,4$, a směrovými vektory ${}_i\mathbf{n}_{Dy}^G$, $i=1,2,\dots,4$. Vzdálenosti označme ${}_id$ a potom

$${}_id = \frac{|\mathbf{u} \times {}_i\mathbf{n}_{Dy}^G|}{|{}_i\mathbf{n}_{Dy}^G|}, \quad i=1,2,\dots,4, \quad (4.9)$$

kde

$${}_i\mathbf{u} = [{}_ix_T^G - {}_ix_D^G, {}_iy_T^G - {}_iy_D^G, {}_iz_T^G - {}_iz_D^G]^T.$$

Úhlové rychlosti otáčení kol jsou potom dány vztahem

$${}_i\omega = \frac{|{}_i\mathbf{v}_D^G|}{{}_id}, \quad i = 1, 2, \dots, 4. \quad (4.10)$$

Otáčení kol bude zajištěno elektromotory, které budou buzeny tak, aby jejich frekvence otáčení odpovídala ${}_i\omega$. Výchylka ${}_i\varphi_{Ey}(t)$ musí tedy odpovídat vztahu

$${}_i\varphi_{Ey}(t) = ({}_i\omega)t, \quad i = 1, 2, \dots, 4. \quad (4.11)$$

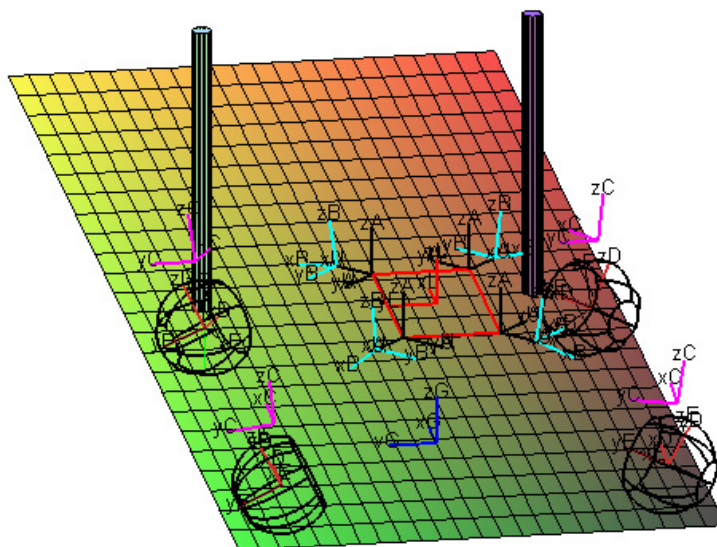
Výsledkem jsou rovnice (4.8) a (4.10), pomocí kterých jsme schopni při nastavených hodnotách volitelných parametrů podvozku určit úhly ${}_i\varphi_{Cz}$, $i = 1, 2, \dots, 4$ a úhlové rychlosti ${}_i\omega$, $i = 1, 2, \dots, 4$, tak, aby se podvozek pohyboval požadovaným směrem pouze v závislosti na R a v . Rovnice (4.8) a (4.10) jsou závislé pouze na poloměru požadované zatáčky a na rychlosti jejího průjezdu, proto jsou vhodné pro řízení skutečného modelu, kde právě tyto dva parametry bude uživatel ovládat.

Zde si můžete prohlédnout animace zatáček:

- [konstantní poloměr](#)
- [proměnný poloměr](#)

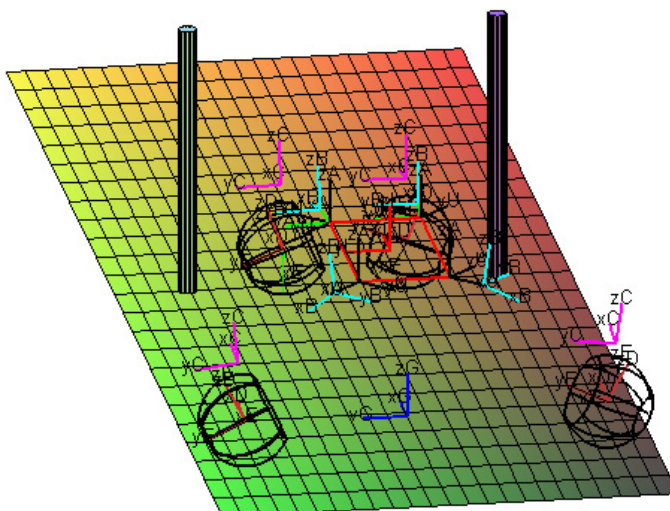
4.2 Průjezd zúženým místem

Při provozu robotizovaného podvozku, ať už v urbanizovaném prostředí nebo ve volné přírodě, může dojít k tomu, že podvozek díky svému rozchodu nedokáže projet zúženým místem. Takové místo mohou představovat třeba zárubně dveří, které v mnoha případech bývají užší než rozchod běžně prodávaných invalidních vozíků. V přírodě se těchto míst vyskytuje daleko více, např. úzké místo tvořené dvěma stromy, úzká lávka, apod. Takový případ je také znázorněn na Obr.4.3.



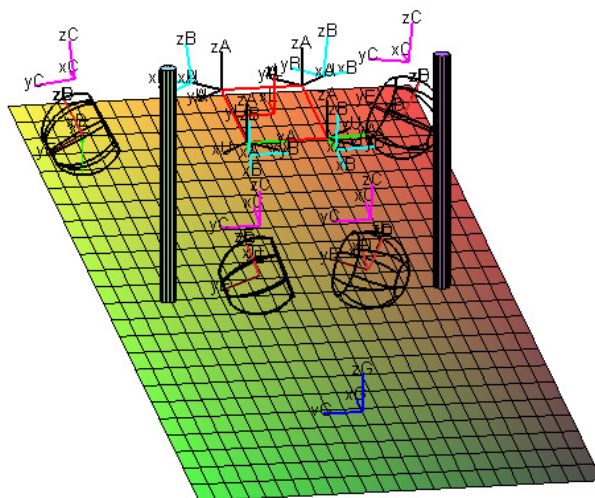
Obr.4.3: Průjezd zúženým místem I

Aby mohl podvozek takovýmto místem projet a zároveň si zachovat co největší stabilitu, musí nejprve pootočit přední nohy o příslušný úhel φ_{Az} (úhel rejdu) tak, aby došlo k přiblížení kol, jak je znázorněno na následujícím obrázku (Obr.4.4). V takovéto konfiguraci projede předními koly zúžené místo a kola se následně vrací do původní pozice.



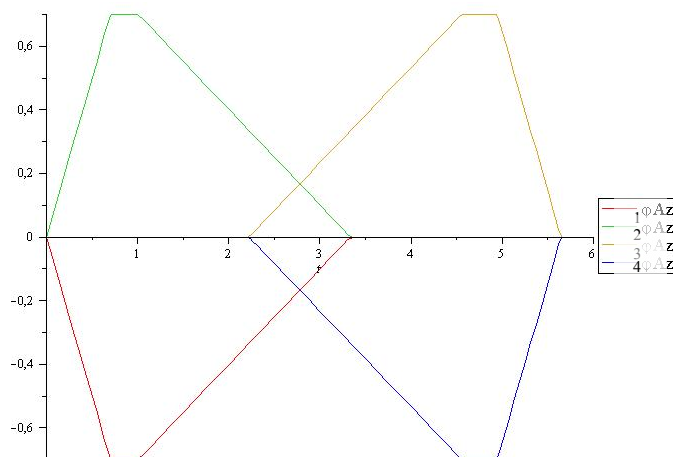
Obr.4.4: Průjezd zúženým místem II

Celý postup se poté opakuje i pro zadní kola, viz Obr.4.5, a po projetí zúženého místa se podvozek vrátí do původní konfigurace.



Obr.4.5: Průjezd zúženým místem III

Na následujícím grafu je znázorněno, jak se mění úhly rejdů při výše uvedeném manévru.



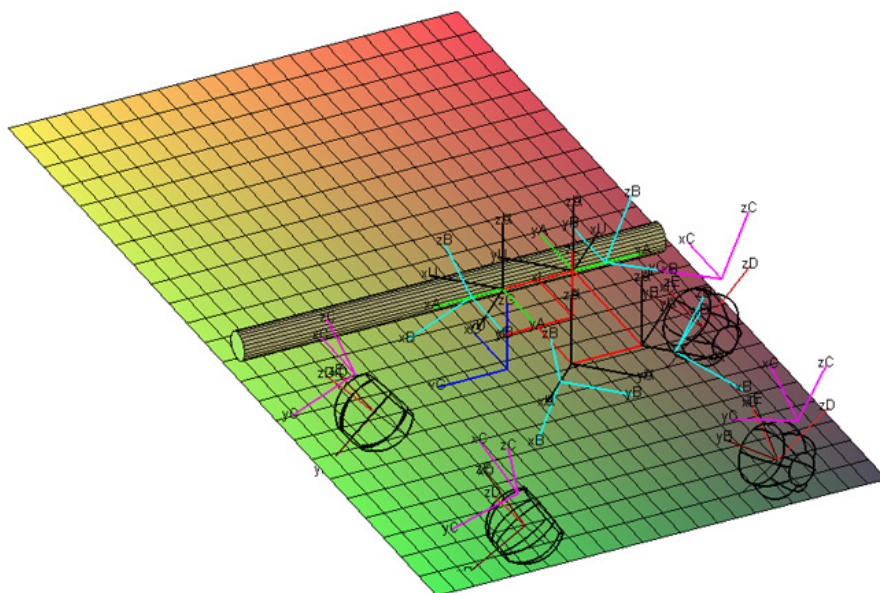
Graf 4.1: Změna úhlu $i \varphi_{Az}$ v čase

[Animace průjezdu zúženým místem.](#)

4.3 Překonání překážky překročením

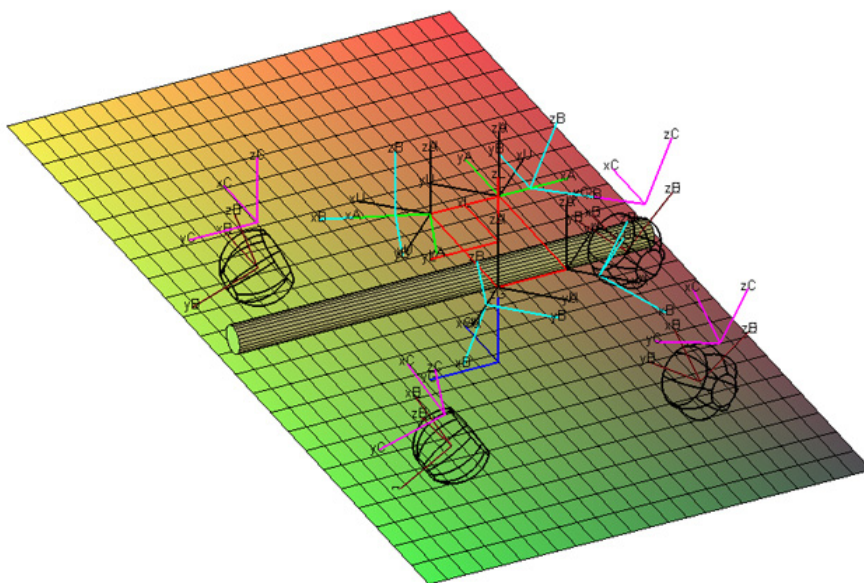
Daleko náročnější manévr, než je průjezd zúženým místem, je překonání ležící překážky překročením. Tuto překážku může představovat ležící kmen stromu, obrubník nebo jiná překážka, jejíž velikost je větší než poloměr kol. Překážky o velikosti menší než je poloměr kola, by měl být podvozek schopen překonat sám. Při najetí kola na takovouto překážku by sama regulační soustava měla upravit nastavení podvozku tak, aby podvozek překážku bez problémů překonal.

Dále se tedy budeme zabývat překážkou větší než je poloměr kola. Na následujících obrázcích jsou znázorněny jednotlivé fáze tohoto manévru.

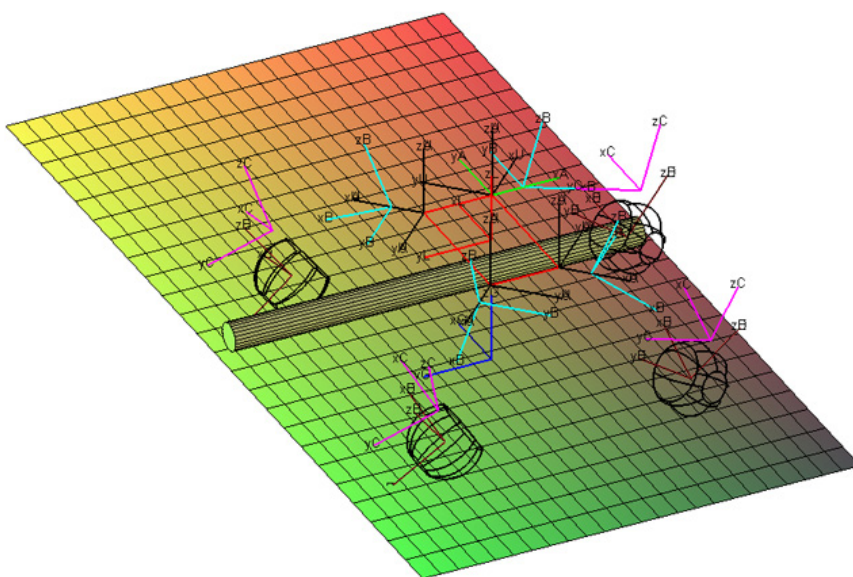


Obr.4.6: Příjezd k překážce

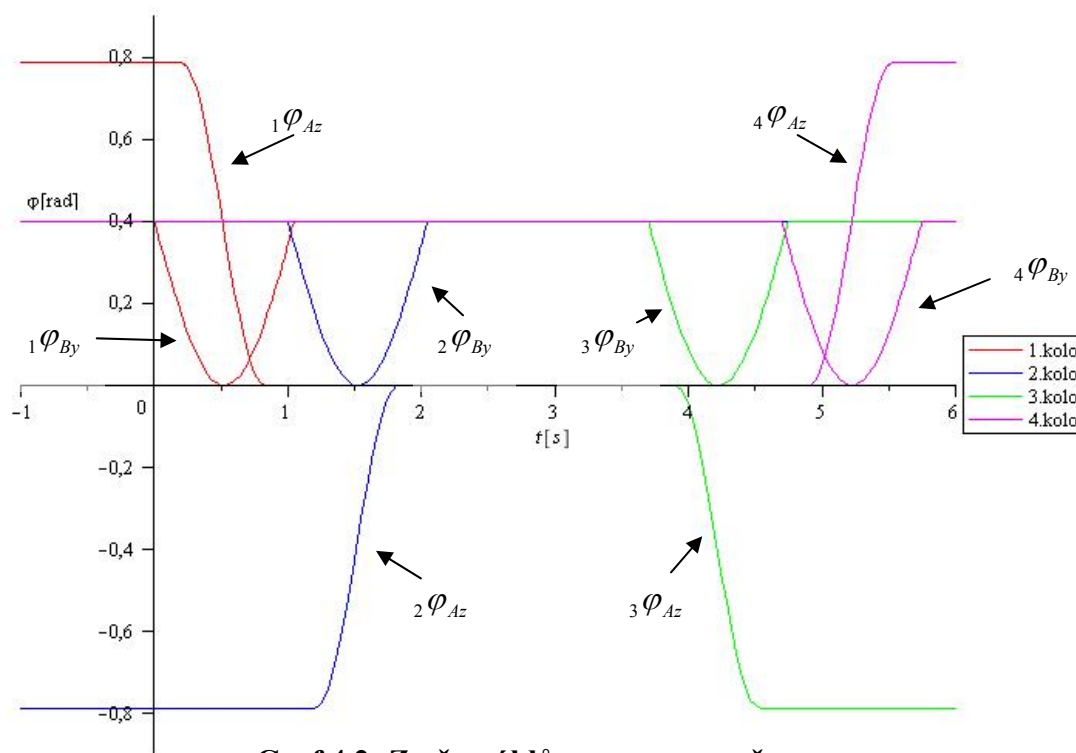
Podvozek se k překážce přiblíží v takové konfiguraci, kdy mají přední kola největší rozchod a zadní kola jsou v základní poloze, viz Obr.4.6. Potom postupně jednotlivými koly překoná překážku. Toho je docíleno současnou změnou úhlů ${}_i\varphi_{Az}$ a ${}_i\varphi_{By}$, jak je znázorněno v Grafu 4.2. Na Obr.4.7 a Obr.4.8 je ve dvou krocích naznačeno překonání překážky prvním kolem.



Obr.4.7: Překonávání překážky 1. kolem



Obr.4.8: Dokončení překonávání překážky 1. kolem



Je zřejmé, že změny úhlů $i\varphi_{Az}$ a $i\varphi_{By}$ znázorněné v grafu 4.2 nejsou univerzální pro překonání obecné překážky. Z toho důvodu budou změny úhlů $i\varphi_{Az}$ a $i\varphi_{By}$ modifikovány parametry, které budou zjištěny např. z čidel nebo zadané přímo uživatelem.

[Překročení překážky překročením.](#)

4.4 Pohyb do schodů

Nejnáročnějším manévrem robotizovaného podvozku bude nepochybně chůze do schodů. V běžném životě se setkáváme se schody rozličných parametrů - výrazně se mění úhel sklonu schodů, šířka a výška schodu, počet schodů. Všechny tyto variability značně ztěžují překonání této běžné překážky.

Podvozek, který je řešen v této práci, je samozřejmě tohoto složitého manévru schopen, ale samostatné řízení manévru, které musí být dostatečně variabilní, aby pokrylo všechny druhy schodů, bude velice složité. Podvozek, stejně jako v předchozím případě, musí být osazen čidly, které odměří parametry schodiště, podle kterých se následně změní řídicí algoritmus.

5 Závěr

Tato diplomová práce obsahuje 3D matematický model robotizovaného podvozku. Matematický model bude v budoucnu doplněn o úlohu zrychlení, což, jak doufám, bude obsahem mé další práce na tomto projektu.

Pohyb vozidla reprezentovaného matematickým modelem je odvozen od otáčení dvou kol, jejichž úhlová rychlost se počítá z předepsaného poloměru zatáčky a z požadované rychlosti pohybu vozidla. Pohyb vozidla odvozený od pohybu pouhých dvou kol je v mírném rozporu s realitou. Je to však nutná daň za zjednodušení reálné, tedy dynamické úlohy, na úlohu kinematickou. Po usazení všech kol na pojezdovou plochu zbývá podvozku pět stupňů volnosti. Dva stupně volnosti podvozku odebírá předpoklad automatické regulace vodorovné polohy, tedy předpoklad $\varphi_x = \varphi_y = 0$. Zbýlé stupně volnosti φ_z, x_L^G, y_L^G odebereme výše zmíněnými podmínkami valení.

Animace základních manévřů uvedených v kapitole 4, jako je jízda do zatáčky, jízda do zatáčky s proměnným poloměrem, průjezd zúženým místem, překonání překážky překročením, jsou zveřejněny na uvedených internetových adresách. Nejsložitější manévr je jízda do zatáčky po zvlněné ploše. Náročnost výpočtu dokazuje to, že i na velice výkonném počítači SGI Altix 350 trval výpočet této stosnímkové animace několik dnů. Pro výpočet této animace bylo také nutné naprogramovat vlastní proceduru pro řešení soustavy diferenciálních rovnic podle Runge - Kutta. Funkce *dsolve* použitá v softwaru Maple pro řešení soustav diferenciálních rovnic se v případě takto složitých rovnic neosvědčila.

6 Seznam použité literatury

- [1] Brát, V., Rosenberg, J., Jáč, V.: *Kinematika*, SNTL, Praha 1987
- [2] Juliš, K., Brepta, R. a kol.: *Mechanika I. díl, Statika a mechanika*, SNTL, Praha 1986
- [3] Rektorys, K. a kol.: *Přehled užití matematiky*, SNTL, Praha 1981
- [4] Medůna, O., *Mechanika podvozku se čtyřmi nezávisle zavěšenými koly s tlumiči a pružinami s řízeným předpětím*, Liberec, Diplomová práce, 2002
- [5] Dudek, G, Jenkin, M., *Computational Principles of Mobile Robotics*, Cambridge University Press

7 Seznam příloh

Popsaný program v softwaru Maple, podle kterého probíhal výpočet. (10 stran)

Načtení potřebných knihoven

```
restart:
with(linalg):
with(plots):
with(plottools):with(DEtools):
```

Procedury

procedura, která převede vektor o čtyřech složkách na vektor o třech složkách

```
cnvrt:=proc(r::vector(4))
```

```
local c;
```

```
c:=convert(r,list);
```

```
return([c[1],c[2],c[3]]) end;
```

procedura, která po zadání rozměrových parametrů kola kolo vykreslí

```
draw_wheel:=proc(R,h1,h2,c,t)
```

```
local u2,c1,a,ta,tc,u1,c2;
```

```
u1:=arcsin(h1/2/R);
```

```
u2:=arcsin(h2/2/R);
```

```
c1:=circle([0,0],R*cos(u1));
```

```
c2:=circle([0,0],R*cos(u2));
```

```
a:=arc([0,0],R,-u1..u2);
```

```
tc[1]:=transform((x,y)->[x,-h1/2,y]);
```

```
tc[2]:=transform((x,y)->[x,h2/2,y]);
```

```
ta[0]:=transform((x,y)->[x,y,0]);
```

```
ta[1]:=transform((x,y)->[x*cos(Pi/4),y,-x*sin(Pi/4)]);
```

```
ta[2]:=transform((x,y)->[x*cos(2*Pi/4),y,-x*sin(2*Pi/4)]);
```

```
ta[3]:=transform((x,y)->[x*cos(3*Pi/4),y,-x*sin(3*Pi/4)]);
```

```
ta[4]:=transform((x,y)->[-x,y,0]);
```

```
ta[5]:=transform((x,y)->[x*cos(5*Pi/4),y,-x*sin(5*Pi/4)]);
```

```
ta[6]:=transform((x,y)->[x*cos(6*Pi/4),y,-x*sin(6*Pi/4)]);
```

```
ta[7]:=transform((x,y)->[x*cos(7*Pi/4),y,-x*sin(7*Pi/4)]);
```

```
display({tc[1](c1),tc[2](c2),
```

```
ta[0](a),ta[1](a),ta[2](a),ta[3](a),ta[4](a),ta[5](a),ta[6](a),ta[7](a)},
```

```
scaling=constrained,color=c,thickness=t)
```

```
end;
```

procedura, která slouží k vykreslení souřadnicového systému reprezentovaného transformační maticí **T**

```
S:=proc(T::matrix(4,4),c,t,L::list)
```

```
local l,o;
```

```
l:=[T[1,4],T[2,4],T[3,4]]; 
```

```
o[1]:=cnvrt(evalm(T&*vector(4,[1,0,0,1])));
```

```
o[2]:=cnvrt(evalm(T&*vector(4,[0,1,0,1])));
```

```
o[3]:=cnvrt(evalm(T&*vector(4,[0,0,1,1])));
```

```
{display(line(l,o[1]),line(l,o[2]),line(l,o[3]),color=c,thickness=t),
```

```
display(textplot3d([op(o[1]),L[1]],textplot3d([op(o[2]),L[2]]),
```

```
textplot3d([op(o[3]),L[3]]),color=black)}
```

```
end;
```

procedura, která vykreslí v souřadnicovém systému s transformační maticí **T** obdélník o zadaných rozměrech

```
Ob:=proc(T::matrix(4,4),a,b,c,t)
```

```
local l,h;
```

```
l[1]:=cnvrt(evalm(T&*vector(4,[a/2,b/2,0,1])));
```

```
l[2]:=cnvrt(evalm(T&*vector(4,[-a/2,b/2,0,1])));
```

```
l[3]:=cnvrt(evalm(T&*vector(4,[-a/2,-b/2,0,1])));
```

```
l[4]:=cnvrt(evalm(T&*vector(4,[a/2,-b/2,0,1])));
```

```
h[1]:=line(l[1],l[2]);
```

```
h[2]:=line(l[2],l[3]);
```

```
h[3]:=line(l[3],l[4]);
```

```
h[4]:=line(l[4],l[1]);
```

```
display(h[1],h[2],h[3],h[4],color=c,thickness=t) end;
```

procedura pro transformaci kola do požadované polohy

```
tr:=transform((x,y,z)->cnvrt(evalm(TGE[j]&*vector(4,[x,y,z,1]))));
```

procedura pro vytvoření matice, která reprezentuje natočení souřadnicového systému okolo všech souřadnicových os podle modelu RPY a posunutí počátku o **dx,dy,dz**

```
RPY:=proc(phi_x,phi_y,phi_z,dx,dy,dz)
```

```
local T1,T2,T3,T;
```

```
T3:=matrix(4,4,[cos(phi_z),-sin(phi_z),0,0,sin(phi_z),cos(phi_z),0,0,0,0,1,0,0,0,0,0]);
```

```
T2:=matrix(4,4,[cos(phi_y),0,sin(phi_y),0,0,1,0,0,-sin(phi_y),0,cos(phi_y),0,0,0,0,0]);
```

```
T1:=matrix(4,4,[1,0,0,0,0,cos(phi_x),-sin(phi_x),0,0,sin(phi_x),cos(phi_x),0,0,0,0,0]);
```

```
T:=evalm(T3&*T2&*T1);
```

```
T[1,4]:=dx;T[2,4]:=dy;T[3,4]:=dz;T[4,4]:=1;
```

```

evalm(T);end proc:
procedura pro vytvoření matice, která reprezentuje otočení okolo osy x a posunutí počátku o dx,dy,dz
rotx:=proc(u,dx,dy,dz)
matrix(4,4,[1,0,0,dx,0,cos(u),-sin(u),dy,0,sin(u),cos(u),dz,0,0,0,1]) end:
procedura pro vytvoření matice, která reprezentuje otočení okolo osy y a posunutí počátku o dx,dy,dz
roty:=proc(u,dx,dy,dz)
matrix(4,4,[cos(u),0,sin(u),dx,0,1,0,dy,-sin(u),0,cos(u),dz,0,0,0,1]) end:
procedura pro vytvoření matice, která reprezentuje otočení okolo osy z a posunutí počátku o dx,dy,dz
rotz:=proc(u,dx,dy,dz)
matrix(4,4,[cos(u),-sin(u),0,dx,sin(u),cos(u),0,dy,0,0,1,dz,0,0,0,1]) end:
procedura pro vytvoření ekvidistantní plochy k ploše  $f(x,y)$ 
eq:=proc(xE,yE)
local x0,y0,zE,bod_d,n_0,stred;
bod_d:=vector(3,[x0,y0,f(x0,y0)]);
n_0:=normalize(vector(3,[-eval(diff(f(x,y),x),{x=x0,y=y0}),-eval(diff(f(x,y),y),{x=x0,y=x0}),1]));
stred:=evalm(bod_d+R*n_0);
fsolve({stred[1]-xE,stred[2]-yE,stred[3]-zE},{x0,y0,zE});assign(%);
zE;end proc:

```

Rozšířené transformační matice

```

matice, která reprezentuje základní rám
T:=matrix(4,4,[1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1]);
transformace z L do G (RPY)
TGL:=RPY(phi_x(t),phi_y(t),phi_z,LX,LY,L[z](t));
transformace z U[i] do L (RPY)  $T^{LU}$ 
TLU[1]:=evalm(RPY(0,0,phi_Uz[1],Ux[1],Uy[1],Uz[1]));
TLU[2]:=evalm(RPY(0,0,phi_Uz[2],Ux[2],Uy[2],Uz[2]));
TLU[3]:=evalm(RPY(0,0,phi_Uz[3],Ux[3],Uy[3],Uz[3]));
TLU[4]:=evalm(RPY(0,0,phi_Uz[4],Ux[4],Uy[4],Uz[4]));
unassign('i');
trasformace z U[i] do G  $T^{GU}$ 
TGU[1]:=evalm(TGL&*TLU[1]);
TGU[2]:=evalm(TGL&*TLU[2]);
TGU[3]:=evalm(TGL&*TLU[3]);
TGU[4]:=evalm(TGL&*TLU[4]);
transformace A[i] do U[i] (rejd)  $T^{UA}$ 
TUA[1]:=evalm(rotz(phi_Az[1](t),Ax[1],Ay[1],Az[1]));
TUA[2]:=evalm(rotz(phi_Az[2](t),Ax[2],Ay[1],Az[2]));
TUA[3]:=evalm(rotz(phi_Az[3](t),Ax[3],Ay[3],Az[3]));
TUA[4]:=evalm(rotz(phi_Az[4](t),Ax[4],Ay[4],Az[4]));
transformace z A[i] do G  $T^{GA}$ 
TGA[1]:=evalm(TGL&*TLU[1]&*TUA[1]);
TGA[2]:=evalm(TGL&*TLU[2]&*TUA[2]);
TGA[3]:=evalm(TGL&*TLU[3]&*TUA[3]);
TGA[4]:=evalm(TGL&*TLU[4]&*TUA[4]);
transformace z B[i] do A[i] (RPY)  $T^{AB} \phi_{By}$ 
TAB[1]:=evalm(roty(phi_By1,Bx[1],By[1],Bz[1]));
TAB[2]:=evalm(roty(phi_By2,Bx[2],By[2],Bz[2]));
TAB[3]:=evalm(roty(phi_By3,Bx[3],By[3],Bz[3]));
TAB[4]:=evalm(roty(phi_By4,Bx[4],By[4],Bz[4]));
unassign('i');
transformace z B[i] do G  $T^{GB}$ 
TGB[1]:=evalm(TGL&*TLU[1]&*TUA[1]&*TAB[1]);
TGB[2]:=evalm(TGL&*TLU[2]&*TUA[2]&*TAB[2]);
TGB[3]:=evalm(TGL&*TLU[3]&*TUA[3]&*TAB[3]);
TGB[4]:=evalm(TGL&*TLU[4]&*TUA[4]&*TAB[4]);
unassign('i');
transformace z C[i] do B[i] (rotace okolo zB, pivotace)  $T^{BC}$ 
TBC[1]:=evalm(rotz(phi_Cz1,Cx[1],Cy[1],Cz[1]));
TBC[2]:=evalm(rotz(phi_Cz2,Cx[2],Cy[2],Cz[2]));
TBC[3]:=evalm(rotz(phi_Cz3,Cx[3],Cy[3],Cz[3]));
TBC[4]:=evalm(rotz(phi_Cz4,Cx[4],Cy[4],Cz[4]));
unassign('i');
transformace z B[i] do G  $T^{GB}$ 
TGC[1]:=evalm(TGL&*TLU[1]&*TUA[1]&*TAB[1]&*TBC[1]);

```

```
TGC[2]:=evalm(TGL&*TLU[2]&*TUA[2]&*TAB[2]&*TBC[2]);
TGC[3]:=evalm(TGL&*TLU[3]&*TUA[3]&*TAB[3]&*TBC[3]);
TGC[4]:=evalm(TGL&*TLU[4]&*TUA[4]&*TAB[4]&*TBC[4]);
unassign('i');
```

transformace z **D[i]** do **C[i]** (translace ve směru **zC** a rotace okolo **xC**) T^{CD}

```
TCD[1]:=evalm(rotx(phi_Cx,Dx[1],Dy[1],Dz[1]));
TCD[2]:=evalm(rotx(phi_Cx2,Dx[2],Dy[2],Dz[2]));
TCD[3]:=evalm(rotx(phi_Cx,Dx[3],Dy[3],Dz[3]));
TCD[4]:=evalm(rotx(phi_Cx2,Dx[4],Dy[4],Dz[4]));
unassign('i');
```

transformace z **D[i]** do **G** T^{GD}

```
TGD[1]:=evalm(TGL&*TLU[1]&*TUA[1]&*TAB[1]&*TBC[1]&*TCD[1]);
TGD[2]:=evalm(TGL&*TLU[2]&*TUA[2]&*TAB[2]&*TBC[2]&*TCD[2]);
TGD[3]:=evalm(TGL&*TLU[3]&*TUA[3]&*TAB[3]&*TBC[3]&*TCD[3]);
TGD[4]:=evalm(TGL&*TLU[4]&*TUA[4]&*TAB[4]&*TBC[4]&*TCD[4]);
unassign('i');
```

Zadané hodnoty

```
phi_x:=(t)->0;phi_y:=(t)->0;
```

```
L[z]:=(t)->f(LX,LY)+1.5;
```

```
Ux[1]:=1.;Ux[2]:=Ux[1];Ux[3]:=-Ux[1];Ux[4]:=Ux[3];
Uy[1]:=0.5;Uy[2]:=-Uy[1];Uy[3]:=Uy[1];Uy[4]:=-Uy[3];
Uz[1]:=0;Uz[2]:=Uz[1];Uz[3]:=Uz[1];Uz[4]:=Uz[3];
```

```
Ax[1]:=0;Ax[2]:=Ax[1];Ax[3]:=Ax[1];Ax[4]:=Ax[1];
Ay[1]:=0;Ay[2]:=0;Ay[3]:=0;Ay[4]:=0;
Az[1]:=0;Az[2]:=Az[1];Az[3]:=Az[1];Az[4]:=Az[1];
```

```
Bx[1]:=0.45;Bx[2]:=Bx[1];Bx[3]:=Bx[1];Bx[4]:=Bx[3];
By[1]:=0;By[2]:=-By[1];By[3]:=By[1];By[4]:=-By[3];
Bz[1]:=0;Bz[2]:=Bz[1];Bz[3]:=Bz[1];Bz[4]:=Bz[3];
```

```
phi_Uz[1]:=evalf(Pi/4);phi_Uz[2]:=-phi_Uz[1];phi_Uz[3]:=evalf(Pi/2+phi_Uz[1]);phi_Uz[4]:=-phi_Uz[3];
```

```
phi_Az[1]:=(t)->evalf(0);phi_Az[2]:=(t)->-phi_Az[1](t);phi_Az[3]:=(t)->-phi_Az[1](t);phi_Az[4]:=(t)->phi_Az[1](t);
```

```
Cx[1]:=1.8;Cx[2]:=Cx[1];Cx[3]:=Cx[1];Cx[4]:=Cx[3];
Cy[1]:=0;Cy[2]:=Cy[1];Cy[3]:=Cy[1];Cy[4]:=Cy[3];
Cz[1]:=0;Cz[2]:=Cz[1];Cz[3]:=Cz[1];Cz[4]:=Cz[3];
```

```
Dx[1]:=0;Dx[2]:=Dx[1];Dx[3]:=Dx[1];Dx[4]:=Dx[3];
Dy[1]:=0;Dy[2]:=-Dy[1];Dy[3]:=Dy[1];Dy[4]:=-Dy[3];
Dz[1]:=-0.7;Dz[2]:=Dz[1];Dz[3]:=Dz[1];Dz[4]:=Dz[3];
```

```
phi_Cx:=evalf(-Pi/180*20);
phi_Cx2:=-phi_Cx;
```

```
R:=0.5;    poloměr kola
```

```
phi_E[1]:=(t)->omega1*t;
phi_E[2]:=(t)->omega2*t;
phi_E[3]:=(t)->omega3*t;
phi_E[4]:=(t)->omega4*t;
```

Výpočet potřebného natočení kol o úhel ϕ_{Cz_i} a úhlové rychlosti pro průjezd zatáčkou daného poloměru r požadovanou rychlostí v

definice pomocného úhlu phiz, který slouží k výpočtu Omega, což je úhlová rychlost průjezdu zatáčkou

```
phiz:=-Omega*t;
```

výpočet úhlové rychlosti průjezdu zatáčkou

```
Omega:=v/r;
```

určení x -ové a y -ové souřadnice bodu **L** - uvažujeme, že při zatáčení se podvozek pohybuje po kružnici se středem ve středu zatáčky

```
L[x]:=r*sin(-phiz);
```

```
L[y]:=(r-r*cos(-phiz));
```

výpočet pomocné transformační matice z **L** do **G**

```
TGL_p:=rotz(phiz,L[x],L[y],L[z](t));
```

výpočet pomocné transformační matice z **D** do **G**

```
TGD_p[1]:=evalm(TGL_p&*TLU[1]&*TUA[1]&*TAB[1]&*TBC[1]&*TCD[1]);
TGD_p[2]:=evalm(TGL_p&*TLU[2]&*TUA[2]&*TAB[2]&*TBC[2]&*TCD[2]);
TGD_p[3]:=evalm(TGL_p&*TLU[3]&*TUA[3]&*TAB[3]&*TBC[3]&*TCD[3]);
TGD_p[4]:=evalm(TGL_p&*TLU[4]&*TUA[4]&*TAB[4]&*TBC[4]&*TCD[4]);
```

výpočet úhlových rychlostí a natočení se bude provádět pro případ rovné pojezdové plochy

```
f:=(x,y)->0:f_x:=diff(f(x,y),x):f_y:=diff(f(x,y),y):f_dx:=unapply(f_x,[x,y]):f_dy:=unapply(f_y,[x,y]):
```

rozšířený vektor, který definuje polohu bodu dotyku kola a plochy f

```
rT[1]:=vector(4,[x01,y01,f(x01,y01),1]);
rT[2]:=vector(4,[x02,y02,f(x02,y02),1]);
rT[3]:=vector(4,[x03,y03,f(x03,y03),1]);
rT[4]:=vector(4,[x04,y04,f(x04,y04),1]);
```

definice normálového vektoru k ploše f v bodě $[x0,y0]$

```
n_0[1]:=Vector(3,[-f_dx(x01,y01)/sqrt(f_dx(x01,y01)^2+f_dy(x01,y01)^2+1^2),-
f_dy(x01,y01)/sqrt(f_dx(x01,y01)^2+f_dy(x01,y01)^2+1^2),1/sqrt(f_dx(x01,y01)^2+f_dy(x01,y01)^2+1^2)]);
```

```
n_0[2]:=Vector(3,[-f_dx(x02,y02)/sqrt(f_dx(x02,y02)^2+f_dy(x02,y02)^2+1^2),-
f_dy(x02,y02)/sqrt(f_dx(x02,y02)^2+f_dy(x02,y02)^2+1^2),1/sqrt(f_dx(x02,y02)^2+f_dy(x02,y02)^2+1^2)]);
```

```
n_0[3]:=Vector(3,[-f_dx(x03,y03)/sqrt(f_dx(x03,y03)^2+f_dy(x03,y03)^2+1^2),-
f_dy(x03,y03)/sqrt(f_dx(x03,y03)^2+f_dy(x03,y03)^2+1^2),1/sqrt(f_dx(x03,y03)^2+f_dy(x03,y03)^2+1^2)]);
```

```
n_0[4]:=Vector(3,[-f_dx(x04,y04)/sqrt(f_dx(x04,y04)^2+f_dy(x04,y04)^2+1^2),-
f_dy(x04,y04)/sqrt(f_dx(x04,y04)^2+f_dy(x04,y04)^2+1^2),1/sqrt(f_dx(x04,y04)^2+f_dy(x04,y04)^2+1^2)]);
```

vytvoření vektorové rovnice, která představuje polohu bodu **D** (středu kola), vyjádřenou pomocí vektorů **n_0** a **rT**a pomocí matice **TGD_p**

```
RCE123[1]:=evalm(cnvrt(rT[1])+R*n_0[1]-cnvrt(evalm(TGD_p[1]&*vector(4,[0,0,0,1]))));
RCE123[2]:=evalm(cnvrt(rT[2])+R*n_0[2]-cnvrt(evalm(TGD_p[2]&*vector(4,[0,0,0,1]))));
RCE123[3]:=evalm(cnvrt(rT[3])+R*n_0[3]-cnvrt(evalm(TGD_p[3]&*vector(4,[0,0,0,1]))));
RCE123[4]:=evalm(cnvrt(rT[4])+R*n_0[4]-cnvrt(evalm(TGD_p[4]&*vector(4,[0,0,0,1]))));
```

zadání počátečních hodnot veličin, které známe a následně dopočítání ostatních

```
LY:=0:LX:=0:t:=0:phi_z:=0:
```

```
fsolve({RCE123[1][1],RCE123[1][2],RCE123[1][3]},{phi_By1,x01,y01},phi_By1=-Pi/3..Pi/3):assign(%):
```

```
fsolve({RCE123[2][1],RCE123[2][2],RCE123[2][3]},{phi_By2,x02,y02},phi_By2=-Pi/3..Pi/3):assign(%):
```

```
fsolve({RCE123[3][1],RCE123[3][2],RCE123[3][3]},{phi_By3,x03,y03},phi_By3=-Pi/3..Pi/3):assign(%):
```

```
fsolve({RCE123[4][1],RCE123[4][2],RCE123[4][3]},{phi_By4,x04,y04},phi_By4=-Pi/3..Pi/3):assign(%):
```

odřazení proměnné t

```
unassign('t');
```

výpočet vektorů **rD[i]**, což jsou polohové vektory bodů **D**, vyjádřené v globálním souřadnicovém systému

```
rD[1]:=evalm(TGD_p[1]&*vector(4,[0,0,0,1]));
rD[2]:=evalm(TGD_p[2]&*vector(4,[0,0,0,1]));
rD[3]:=evalm(TGD_p[3]&*vector(4,[0,0,0,1]));
rD[4]:=evalm(TGD_p[4]&*vector(4,[0,0,0,1]));
```

zadání poloměru zatáčky a rychlosti průjezdu

```
v:=2:r:=5:
```

derivace vektorů **rD[i]**, čímž získáme vektory představující rychlost bodu **D**

```
vD[1]:=evalm(map(diff,rD[1],t)):
vD[2]:=evalm(map(diff,rD[2],t)):
vD[3]:=evalm(map(diff,rD[3],t)):
vD[4]:=evalm(map(diff,rD[4],t)):
```

přiřazení hodnoty 0 proměnné t

```
t:=0:
```

výpočet jednotkového vektoru ve směru osy y souřadnicového systému **D**

```
ryC[1]:=cnvrt(evalm(TGD_p[1]&*vector(4,[0,1,0,1])-TGD_p[1]&*vector(4,[0,0,0,1])));
ryC[2]:=cnvrt(evalm(TGD_p[2]&*vector(4,[0,1,0,1])-TGD_p[2]&*vector(4,[0,0,0,1])));
ryC[3]:=cnvrt(evalm(TGD_p[3]&*vector(4,[0,1,0,1])-TGD_p[3]&*vector(4,[0,0,0,1])));
ryC[4]:=cnvrt(evalm(TGD_p[4]&*vector(4,[0,1,0,1])-TGD_p[4]&*vector(4,[0,0,0,1])));
```

-vytvoření rovnic sloužících pro výpočet úhlu ϕ_{Cz_i}

-rovnice vznikne jako skalární součin vektorů ryC_i a vD_i

```
RCE12[1]:=evalm(ryC[1]&*cnvrt(vD[1]));
RCE12[2]:=evalm(ryC[2]&*cnvrt(vD[2]));
RCE12[3]:=evalm(ryC[3]&*cnvrt(vD[3]));
RCE12[4]:=evalm(ryC[4]&*cnvrt(vD[4]));
```

numerické řešení předchozích rovnic v daných mezích

```
fsolve({RCE12[1]},{phi_Cz1},phi_Cz1=-Pi..0):assign(%);
```

```
fsolve({RCE12[2]},{phi_Cz2},phi_Cz2=-Pi/2..Pi):assign(%);
```

```
fsolve({RCE12[3]},{phi_Cz3},phi_Cz3=-Pi..0):assign(%);
```

```
fsolve({RCE12[4]},{phi_Cz4},phi_Cz4=0..Pi):assign(%);
```

výpočet vektoru, který je potřebný k výpočtu poloměru, na kterém se kola otáčí

```
uu[1]:=cnvrt(evalm(rT[1]-rD[1]));
uu[2]:=cnvrt(evalm(rT[2]-rD[2]));
uu[3]:=cnvrt(evalm(rT[3]-rD[3]));
uu[4]:=cnvrt(evalm(rT[4]-rD[4]));
výpočet poloměru, na kterém se otáčejí kola
d[1]:=norm(crossprod(uu[1],ryC[1]),2)/norm(ryC[1],2);
d[2]:=norm(crossprod(uu[2],ryC[2]),2)/norm(ryC[2],2);
d[3]:=norm(crossprod(uu[3],ryC[3]),2)/norm(ryC[3],2);
d[4]:=norm(crossprod(uu[4],ryC[4]),2)/norm(ryC[4],2);
výpočet úhlových rychlostí jednotlivých kol
omega1:=eval(norm(vD[1],2)/d[1]);
omega2:=eval(norm(vD[2],2)/d[2]);
omega3:=eval(norm(vD[3],2)/d[3]);
omega4:=eval(norm(vD[4],2)/d[4]);
```

Sestavení základních rovnic

odřazení proměnných

```
unassign('t','phi_By1','phi_By2','phi_By3','phi_By4','x01','x03','x04','x02','y01','y03','y04','y02','LX','LY','phi_z','f','i');
```

rs1-jednotkový vektor ve směru osy x globálního souřadnicového systému

rs2-jednotkový vektor ve směru osy x^C souřadnicového systému s počátkem v bodě C

```
rs1:=vector(3,[1,0,0]);
rs2:=cnvrt(evalm(TGC[2]&*vector(4,[1,0,0,1])-TGC[2]&*vector(4,[0,0,0,1])));
```

transformace z $E[i]$ do G T^{GE}

for i from 1 to 4 do

```
    TGE[i]:=evalm(TGD[i]&*(roty(phi_E[i](t),0,0,0)));
```

end do:unassign('i');

vytvoření inverzních matic k matici T^{GE} , které jsou potřebné k výpočtu rychlostí bodů dotyku, které by při odvalování měly mít nulovou rychlost

```
TEG[1]:=inverse(TGE[1]);
TEG[2]:=inverse(TGE[2]);
TEG[3]:=inverse(TGE[3]);
TEG[4]:=inverse(TGE[4]);
```

derivace transformační matice T^{GE} podle času, derivace je provedená pro všechny možné proměnné

```
d_TGE[1]:=evalm(map(diff,TGE[1],t)+map(diff,TGE[1],phi_By1)*d_phi_By1+map(diff,TGE[1],LX)*d_LX+map(diff,TGE[1],LY)*d_LY+map(diff,TGE[1],x01)*d_x01+map(diff,TGE[1],y01)*d_y01+map(diff,TGE[1],phi_z)*d_phi_z);
```

```
d_TGE[2]:=evalm(map(diff,TGE[2],t)+map(diff,TGE[2],phi_By2)*d_phi_By2+map(diff,TGE[2],LX)*d_LX+map(diff,TGE[2],LY)*d_LY+map(diff,TGE[2],x02)*d_x02+map(diff,TGE[2],y02)*d_y02+map(diff,TGE[2],phi_z)*d_phi_z);
```

```
d_TGE[3]:=evalm(map(diff,TGE[3],t)+map(diff,TGE[3],phi_By3)*d_phi_By3+map(diff,TGE[3],LX)*d_LX+map(diff,TGE[3],LY)*d_LY+map(diff,TGE[3],x03)*d_x03+map(diff,TGE[3],y03)*d_y03+map(diff,TGE[3],phi_z)*d_phi_z);
```

```
d_TGE[4]:=evalm(map(diff,TGE[4],t)+map(diff,TGE[4],phi_By3)*d_phi_By3+map(diff,TGE[4],LX)*d_LX+map(diff,TGE[4],LY)*d_LY+map(diff,TGE[4],x04)*d_x04+map(diff,TGE[4],y04)*d_y04+map(diff,TGE[4],phi_z)*d_phi_z);
```

definice plochy a derivací plochy v bodě, pomocí kterých je definována normála k ploše f v bodě $[x0,y0]$

```
f:=(x,y)>0.15*sin(2.5*x)*cos(2.5*y);f_x:=diff(f(x,y),x):f_y:=diff(f(x,y),y):f_dx:=unapply(f_x,[x,y]):f_dy:=unapply(f_y,[x,y]):
```

rozšířený vektor, který definuje polohu bodu dotyku kola a plochy f

```
rT[1]:=vector(4,[x01,y01,f(x01,y01),1]);
rT[2]:=vector(4,[x02,y02,f(x02,y02),1]);
rT[3]:=vector(4,[x03,y03,f(x03,y03),1]);
rT[4]:=vector(4,[x04,y04,f(x04,y04),1]);
```

definice normálového vektoru k ploše f v bodě $[x0,y0]$

```
n_0[1]:=Vector(3,[-f_dx(x01,y01)/sqrt(f_dx(x01,y01)^2+f_dy(x01,y01)^2+1^2),-f_dy(x01,y01)/sqrt(f_dx(x01,y01)^2+f_dy(x01,y01)^2+1^2),1/sqrt(f_dx(x01,y01)^2+f_dy(x01,y01)^2+1^2)]);
```

```
n_0[2]:=Vector(3,[-f_dx(x02,y02)/sqrt(f_dx(x02,y02)^2+f_dy(x02,y02)^2+1^2),-f_dy(x02,y02)/sqrt(f_dx(x02,y02)^2+f_dy(x02,y02)^2+1^2),1/sqrt(f_dx(x02,y02)^2+f_dy(x02,y02)^2+1^2)]);
```

```
n_0[3]:=Vector(3,[-f_dx(x03,y03)/sqrt(f_dx(x03,y03)^2+f_dy(x03,y03)^2+1^2),-f_dy(x03,y03)/sqrt(f_dx(x03,y03)^2+f_dy(x03,y03)^2+1^2),1/sqrt(f_dx(x03,y03)^2+f_dy(x03,y03)^2+1^2)]);
```

```
n_0[4]:=Vector(3,[-f_dx(x04,y04)/sqrt(f_dx(x04,y04)^2+f_dy(x04,y04)^2+1^2),-f_dy(x04,y04)/sqrt(f_dx(x04,y04)^2+f_dy(x04,y04)^2+1^2),1/sqrt(f_dx(x04,y04)^2+f_dy(x04,y04)^2+1^2)]);
```

vytvoření vektorové rovnice, která představuje polohu bodu D (středu kola) a to vyjádřenou pomocí vektorů n_0 a rT a pomocí matice T^{GD}

```
RCE123[1]:=evalm(cnvrt(rT[1])+R*n_0[1]-cnvrt(evalm(TGD[1]&*vector(4,[0,0,0,1]))));
RCE123[2]:=evalm(cnvrt(rT[2])+R*n_0[2]-cnvrt(evalm(TGD[2]&*vector(4,[0,0,0,1]))));
```



```

RCE123[3]:=evalm(cnvrt(rT[3])+R*n_0[3]-cnvrt(evalm(TGD[3]&*vector(4,[0,0,0,1])))):
RCE123[4]:=evalm(cnvrt(rT[4])+R*n_0[4]-cnvrt(evalm(TGD[4]&*vector(4,[0,0,0,1])))):
derivate RCE123 podle času
d_RCE123[1]:=evalm(map(diff,RCE123[1],t)+map(diff,RCE123[1],phi_By1)*d_phi_By1+map(diff,RCE123[1],LX)*d_LX+map(diff,RCE123[1],LY)*d_LY+map(diff,RCE123[1],x01)*d_x01+map(diff,RCE123[1],y01)*d_y01+map(diff,RCE123[1],phi_z)*d_phi_z):

d_RCE123[2]:=evalm(map(diff,RCE123[2],t)+map(diff,RCE123[2],phi_By2)*d_phi_By2+map(diff,RCE123[2],LX)*d_LX+map(diff,RCE123[2],LY)*d_LY+map(diff,RCE123[2],x02)*d_x02+map(diff,RCE123[2],y02)*d_y02+map(diff,RCE123[2],phi_z)*d_phi_z):

d_RCE123[3]:=evalm(map(diff,RCE123[3],t)+map(diff,RCE123[3],phi_By3)*d_phi_By3+map(diff,RCE123[3],LX)*d_LX+map(diff,RCE123[3],LY)*d_LY+map(diff,RCE123[3],phi_z)*d_phi_z+map(diff,RCE123[3],x03)*d_x03+map(diff,RCE123[3],y03)*d_y03):

d_RCE123[4]:=evalm(map(diff,RCE123[4],t)+map(diff,RCE123[4],phi_By4)*d_phi_By4+map(diff,RCE123[4],LX)*d_LX+map(diff,RCE123[4],LY)*d_LY+map(diff,RCE123[4],phi_z)*d_phi_z+map(diff,RCE123[4],x04)*d_x04+map(diff,RCE123[4],y04)*d_y04):
vykreslení kol
for j from 1 by 2 to 4 do
    kolo[j]:=tr(draw_wheel(R,0.6,0.9,black,2));
end do:

for j from 2 by 2 to 4 do
    kolo[j]:=tr(draw_wheel(R,0.9,0.6,black,2));
end do:
výpočet rychlosti bodu dotyku, jako by byl součástí souřadného systému E
VT[1]:=evalm(d_TGE[1]&*TEG[1]&*rT[1]):
VT[2]:=evalm(d_TGE[2]&*TEG[2]&*rT[2]):

```

```

výpočet složky rychlosti dotykového bodu ve směru vektorů rs1 a rs2
VTrs[1]:=evalm(cnvrt(VT[1])&*rs1):
VTrs[2]:=evalm(cnvrt(VT[2])&*rs2):
VTrs[1]:=simplify(VTrs[1]):
VTrs[2]:=simplify(VTrs[2]):
otočení vektoru rs1 o 90° proti směru hodinových ručiček, výsledkem je vektor krs1
krs1:=evalm(matrix(3,3,[cos(Pi/2),-sin(Pi/2),0,sin(Pi/2),cos(Pi/2),0,0,0,1])&*rs1):
výpočet složky rychlosti dotykového bodu ve směru vektoru krs1
VTkrs[1]:=evalm(cnvrt(VT[1])&*krs1):
VTkrs[1]:=simplify(VTkrs[1]):

```

Řešení počátečních podmínek pro soustavu dif. rovnic

```

odřazení některých proměnných
unassign('phi_By1','phi_By2','x01','x02','y01','y02');
přiřazení hodnot proměnným, které mohou zvolit
y01:=3.4:LX:=0:t:=0:phi_z:=0:
výpočet ostatních hodnot
fsolve({RCE123[1][1],RCE123[1][2],RCE123[1][3]},{phi_By1,x01,LY},phi_By1=-Pi/3..Pi/3);assign(%);
fsolve({RCE123[2][1],RCE123[2][2],RCE123[2][3]},{phi_By2,x02,y02},phi_By2=-Pi/3..Pi/3);assign(%);

```

```

přiřazení počátečních podmínek
phi_By1_0:=phi_By1:phi_By2_0:=phi_By2:phi_z_0:=phi_z:LX_0:=LX:LY_0:=LY:x01_0:=x01:x02_0:=x02:y01_0:=y01:y02_0:=y02:
unassign('phi_By1','phi_By2','x01','x02','LY','y01','y02','LX','t','phi_z');

```

Generování matice soustavy a převod soustavy do kanonického tvaru

```

tvorba matice soustavy a vektoru pravé strany
AA:=genmatrix([d_RCE123[1][1],d_RCE123[1][2],d_RCE123[1][3],d_RCE123[2][1],d_RCE123[2][2],d_RCE123[2][3],VTrs[1],VTrs[2],
VTkrs[1]],[d_phi_By1,d_phi_By2,d_LX,d_LY,d_phi_z,d_x01,d_x02,d_y01,d_y02],prst):
uložení matice soustavy jako jiné matice
BB:=evalm(AA):
-procedura, která slouží k inverzi matice soustavy
-využívá toho, že matice obsahuje spoustu nul
inverzeAA:=proc(M)
    local i,j,nul,C,invC;
    for i from 1 to 9 do
        for j from 1 to 9 do
            if length(M[i,j])<>0 then
                unassign('M[i,j]');
            end if;
        end do;
    end do;
    nul:=matrix(9,9,0);

```

```

    for i from 1 to 9 do nul[i,i]:=1 end do:
    C:=blockmatrix(1,2,[M,nul]);
    invC:=gaussjord(C):
    submatrix(invC,1..9,10..18):
end proc:
využití výše zmíněné procedury k inverzi matice soustavy
invAA:=inverzeAA(AA):
opětovným přiřazením dosadíme do inverzní matice prvky z matice soustavy
AA:=evalm(BB):
vynásobení vektoru pravé strany soustavy zleva invertovanou maticí soustavy
DRCE:=evalm(invAA&*prst):
převedení výrazů pro jednotlivé derivace na funkce, do kterých se bude dosazovat v metodě Runge-Kutta
jedná se o kanonický tvar soustavy diferenciálních rovnic
f1:=unapply(DRCE[1],[phi_By1,phi_By2,LX,LY,phi_z,x01,x02,y01,y02]):
f2:=unapply(DRCE[2],[phi_By1,phi_By2,LX,LY,phi_z,x01,x02,y01,y02]):
f3:=unapply(DRCE[3],[phi_By1,phi_By2,LX,LY,phi_z,x01,x02,y01,y02]):
f4:=unapply(DRCE[4],[phi_By1,phi_By2,LX,LY,phi_z,x01,x02,y01,y02]):
f5:=unapply(DRCE[5],[phi_By1,phi_By2,LX,LY,phi_z,x01,x02,y01,y02]):
f6:=unapply(DRCE[6],[phi_By1,phi_By2,LX,LY,phi_z,x01,x02,y01,y02]):
f7:=unapply(DRCE[7],[phi_By1,phi_By2,LX,LY,phi_z,x01,x02,y01,y02]):
f8:=unapply(DRCE[8],[phi_By1,phi_By2,LX,LY,phi_z,x01,x02,y01,y02]):
f9:=unapply(DRCE[9],[phi_By1,phi_By2,LX,LY,phi_z,x01,x02,y01,y02]):

```

Řešení metodou Runge-Kutta

definice velikosti kroku a počátečního času

```
h:=0.025;t0:=0;
```

vlastní řešení metodou Runge-Kutta

```
for ii from 1 to 30 do
```

```
  t:=t0+h/2;
```

```

k1_1[ii]:=evalf(f1(phi_By1_0,phi_By2_0,LX_0,LY_0,phi_z_0,x01_0,x02_0,y01_0,y02_0));
k1_2[ii]:=evalf(f2(phi_By1_0,phi_By2_0,LX_0,LY_0,phi_z_0,x01_0,x02_0,y01_0,y02_0));
k1_3[ii]:=evalf(f3(phi_By1_0,phi_By2_0,LX_0,LY_0,phi_z_0,x01_0,x02_0,y01_0,y02_0));
k1_4[ii]:=evalf(f4(phi_By1_0,phi_By2_0,LX_0,LY_0,phi_z_0,x01_0,x02_0,y01_0,y02_0));
k1_5[ii]:=evalf(f5(phi_By1_0,phi_By2_0,LX_0,LY_0,phi_z_0,x01_0,x02_0,y01_0,y02_0));
k1_6[ii]:=evalf(f6(phi_By1_0,phi_By2_0,LX_0,LY_0,phi_z_0,x01_0,x02_0,y01_0,y02_0));
k1_7[ii]:=evalf(f7(phi_By1_0,phi_By2_0,LX_0,LY_0,phi_z_0,x01_0,x02_0,y01_0,y02_0));
k1_8[ii]:=evalf(f8(phi_By1_0,phi_By2_0,LX_0,LY_0,phi_z_0,x01_0,x02_0,y01_0,y02_0));
k1_9[ii]:=evalf(f9(phi_By1_0,phi_By2_0,LX_0,LY_0,phi_z_0,x01_0,x02_0,y01_0,y02_0));

-----

k2_1[ii]:=evalf(f1(phi_By1_0+h/2*k1_1[ii],phi_By2_0+h/2*k1_2[ii],LX_0+h/2*k1_3[ii],LY_0+h/2*k1_4[ii],
  phi_z_0+h/2*k1_5[ii],x01_0+h/2*k1_6[ii],x02_0+h/2*k1_7[ii],y01_0+h/2*k1_8[ii],y02_0+
  h/2*k1_9[ii]));

k2_2[ii]:=evalf(f2(phi_By1_0+h/2*k1_1[ii],phi_By2_0+h/2*k1_2[ii],LX_0+h/2*k1_3[ii],LY_0+h/2*k1_4[ii],
  phi_z_0+h/2*k1_5[ii],x01_0+h/2*k1_6[ii],x02_0+h/2*k1_7[ii],y01_0+h/2*k1_8[ii],y02_0+
  h/2*k1_9[ii]));
k2_3[ii]:=evalf(f3(phi_By1_0+h/2*k1_1[ii],phi_By2_0+h/2*k1_2[ii],LX_0+h/2*k1_3[ii],LY_0+h/2*k1_4[ii],
  phi_z_0+h/2*k1_5[ii],x01_0+h/2*k1_6[ii],x02_0+h/2*k1_7[ii],y01_0+h/2*k1_8[ii],y02_0+
  h/2*k1_9[ii]));
k2_4[ii]:=evalf(f4(phi_By1_0+h/2*k1_1[ii],phi_By2_0+h/2*k1_2[ii],LX_0+h/2*k1_3[ii],LY_0+h/2*k1_4[ii],
  phi_z_0+h/2*k1_5[ii],x01_0+h/2*k1_6[ii],x02_0+h/2*k1_7[ii],y01_0+h/2*k1_8[ii],y02_0+
  h/2*k1_9[ii]));
k2_5[ii]:=evalf(f5(phi_By1_0+h/2*k1_1[ii],phi_By2_0+h/2*k1_2[ii],LX_0+h/2*k1_3[ii],LY_0+h/2*k1_4[ii],
  phi_z_0+h/2*k1_5[ii],x01_0+h/2*k1_6[ii],x02_0+h/2*k1_7[ii],y01_0+h/2*k1_8[ii],y02_0+
  h/2*k1_9[ii]));
k2_6[ii]:=evalf(f6(phi_By1_0+h/2*k1_1[ii],phi_By2_0+h/2*k1_2[ii],LX_0+h/2*k1_3[ii],LY_0+h/2*k1_4[ii],
  phi_z_0+h/2*k1_5[ii],x01_0+h/2*k1_6[ii],x02_0+h/2*k1_7[ii],y01_0+h/2*k1_8[ii],y02_0+
  h/2*k1_9[ii]));
k2_7[ii]:=evalf(f7(phi_By1_0+h/2*k1_1[ii],phi_By2_0+h/2*k1_2[ii],LX_0+h/2*k1_3[ii],LY_0+h/2*k1_4[ii],
  phi_z_0+h/2*k1_5[ii],x01_0+h/2*k1_6[ii],x02_0+h/2*k1_7[ii],y01_0+h/2*k1_8[ii],y02_0+
  h/2*k1_9[ii]));
k2_8[ii]:=evalf(f8(phi_By1_0+h/2*k1_1[ii],phi_By2_0+h/2*k1_2[ii],LX_0+h/2*k1_3[ii],LY_0+h/2*k1_4[ii],
  phi_z_0+h/2*k1_5[ii],x01_0+h/2*k1_6[ii],x02_0+h/2*k1_7[ii],y01_0+h/2*k1_8[ii],y02_0+
  h/2*k1_9[ii]));;
k2_9[ii]:=evalf(f9(phi_By1_0+h/2*k1_1[ii],phi_By2_0+h/2*k1_2[ii],LX_0+h/2*k1_3[ii],LY_0+h/2*k1_4[ii],
  phi_z_0+h/2*k1_5[ii],x01_0+h/2*k1_6[ii],x02_0+h/2*k1_7[ii],y01_0+h/2*k1_8[ii],y02_0+
  h/2*k1_9[ii]));

```

```
k4_1[ii]:=evalf(f1(phi_By1_0+h*k3_1[ii],phi_By2_0+h*k3_2[ii],LX_0+h*k3_3[ii],LY_0+h*k3_4[ii],
phi_z_0+h*k3_5[ii],x01_0+h*k3_6[ii],x02_0+h*k3_7[ii],y01_0+h*k3_8[ii],y02_0+
h*k3_9[ii]));

k4_2[ii]:=evalf(f2(phi_By1_0+h*k3_1[ii],phi_By2_0+h*k3_2[ii],LX_0+h*k3_3[ii],LY_0+h*k3_4[ii],
phi_z_0+h*k3_5[ii],x01_0+h*k3_6[ii],x02_0+h*k3_7[ii],y01_0+h*k3_8[ii],y02_0+
h*k3_9[ii]));

k4_3[ii]:=evalf(f3(phi_By1_0+h*k3_1[ii],phi_By2_0+h*k3_2[ii],LX_0+h*k3_3[ii],LY_0+h*k3_4[ii],
phi_z_0+h*k3_5[ii],x01_0+h*k3_6[ii],x02_0+h*k3_7[ii],y01_0+h*k3_8[ii],y02_0+
h*k3_9[ii]));

k4_4[ii]:=evalf(f4(phi_By1_0+h*k3_1[ii],phi_By2_0+h*k3_2[ii],LX_0+h*k3_3[ii],LY_0+h*k3_4[ii],
phi_z_0+h*k3_5[ii],x01_0+h*k3_6[ii],x02_0+h*k3_7[ii],y01_0+h*k3_8[ii],y02_0+
h*k3_9[ii]));

k4_5[ii]:=evalf(f5(phi_By1_0+h*k3_1[ii],phi_By2_0+h*k3_2[ii],LX_0+h*k3_3[ii],LY_0+h*k3_4[ii],
phi_z_0+h*k3_5[ii],x01_0+h*k3_6[ii],x02_0+h*k3_7[ii],y01_0+h*k3_8[ii],y02_0+
h*k3_9[ii]));

k4_6[ii]:=evalf(f6(phi_By1_0+h*k3_1[ii],phi_By2_0+h*k3_2[ii],LX_0+h*k3_3[ii],LY_0+h*k3_4[ii],
phi_z_0+h*k3_5[ii],x01_0+h*k3_6[ii],x02_0+h*k3_7[ii],y01_0+h*k3_8[ii],y02_0+
h*k3_9[ii]));

k4_7[ii]:=evalf(f7(phi_By1_0+h*k3_1[ii],phi_By2_0+h*k3_2[ii],LX_0+h*k3_3[ii],LY_0+h*k3_4[ii],
phi_z_0+h*k3_5[ii],x01_0+h*k3_6[ii],x02_0+h*k3_7[ii],y01_0+h*k3_8[ii],y02_0+
h*k3_9[ii]));

k4_8[ii]:=evalf(f8(phi_By1_0+h*k3_1[ii],phi_By2_0+h*k3_2[ii],LX_0+h*k3_3[ii],LY_0+h*k3_4[ii],
phi_z_0+h*k3_5[ii],x01_0+h*k3_6[ii],x02_0+h*k3_7[ii],y01_0+h*k3_8[ii],y02_0+
h*k3_9[ii]));;

k4_9[ii]:=evalf(f9(phi_By1_0+h*k3_1[ii],phi_By2_0+h*k3_2[ii],LX_0+h*k3_3[ii],LY_0+h*k3_4[ii],
phi_z_0+h*k3_5[ii],x01_0+h*k3_6[ii],x02_0+h*k3_7[ii],y01_0+h*k3_8[ii],y02_0+
h*k3_9[ii]));;
```

```

r_phi_By1[iii]:=evalf(phi_By1_0+h/6*(k1_1[iii]+2*k2_1[iii]+2*k3_1[iii]+k4_1[iii]));
r_phi_By2[iii]:=evalf(phi_By2_0+h/6*(k1_2[iii]+2*k2_2[iii]+2*k3_2[iii]+k4_2[iii]));
F_LX[iii]:=evalf(LX_0+h/6*(k1_3[iii]+2*k2_3[iii]+2*k3_3[iii]+k4_3[iii]));
r_LY[iii]:=evalf(LY_0+h/6*(k1_4[iii]+2*k2_4[iii]+2*k3_4[iii]+k4_4[iii]));
r_phi_z[iii]:=evalf(phi_z_0+h/6*(k1_5[iii]+2*k2_5[iii]+2*k3_5[iii]+k4_5[iii]));
r_x01[iii]:=evalf(x01_0+h/6*(k1_6[iii]+2*k2_6[iii]+2*k3_6[iii]+k4_6[iii]));
r_x02[iii]:=evalf(x02_0+h/6*(k1_7[iii]+2*k2_7[iii]+2*k3_7[iii]+k4_7[iii]));
r_y01[iii]:=evalf(y01_0+h/6*(k1_8[iii]+2*k2_8[iii]+2*k3_8[iii]+k4_8[iii]));
r_y02[iii]:=evalf(y02_0+h/6*(k1_9[iii]+2*k2_9[iii]+2*k3_9[iii]+k4_9[iii]));

```

```
print(phi_By1_0,phi_By2_0,LX_0,LY_0,phi_z_0,x01_0,x02_0,y01_0,y02_0);
```

```
phi_By1_0:=r_phi_By1[ii];phi_By2_0:=r_phi_By2[ii];
phi_z_0:=r_phi_z[ii];LX_0:=r_LX[ii];LY_0:=r_LY[ii];
x01_0:=r_x01[ii];x02_0:=r_x02[ii];
y01_0:=r_y01[ii];y02_0:=r_y02[ii];
t0:=t+h/2;
```

```
end do;
```

Dopočítání potřebných hodnot a vykreslení jednotlivých snímků animace

```
t:=0;
for ii from 1 to 30 do
```

```
t:=t+0.025;
```

```
unassign('LX','LY','x01','x02','y01','y02','phi_z');
```

```
phi_By1:=r_phi_By1[ii];phi_By2:=r_phi_By2[ii];
LX:=r_LX[ii];LY:=r_LY[ii];
x01:=r_x01[ii];x02:=r_x02[ii];
y01:=r_y01[ii];y02:=r_y02[ii];
phi_z:=r_phi_z[ii];
```

```
unassign('d_x01','d_x02','d_LY','d_LX','d_y01','d_y02','d_phi_z','d_phi_By1','d_phi_By2','phi_By3','x03','y03','d_phi_By3','d_x03','d_y03','phi_By4','x04','y04','d_phi_By4','d_x04','d_y04');
```

výpočet hodnot derivací veličin určených z numerického řešení soustavy diferenciálních rovnic

```
fsolve({d_RCE123[1][1],d_RCE123[1][2],d_RCE123[1][3],d_RCE123[2][1],d_RCE123[2][2],d_RCE123[2][3],VTrs[1],VTkrs[1],VTrs[2]},
{d_x01,d_x02,d_LY,d_LX,d_y01,d_y02,d_phi_z,d_phi_By1,d_phi_By2});assign(%);
```

výpočet parametrů týkajících se 3. kola

```
fsolve({RCE123[3][1],RCE123[3][2],RCE123[3][3],d_RCE123[3][1],d_RCE123[3][2],d_RCE123[3][3]},{phi_By3,x03,y03,d_phi_By3,d_x03,d_y03});assign(%);
```

výpočet parametrů týkajících se 4. kola

```
fsolve({RCE123[4][1],RCE123[4][2],RCE123[4][3],d_RCE123[4][1],d_RCE123[4][2],d_RCE123[4][3]},{phi_By4,x04,y04,d_phi_By4,d_x04,d_y04});assign(%);
```

vykreslení normálového vektoru v bodě dotyku prvního kola

```
vektor:=arrow([x01,y01,f(x01,y01)],evalm(R*n_0[1]),.02,.05,.2,cylindrical_arrow,colour=green);
```

vykreslení pojezdové plochy v určitých mezích

```
plocha:=plot3d(f(x,y),x=evalf(LX-4)..evalf(LX+4),y=evalf(LY-4)..evalf(LY+4),scaling=constrained);
```

vykreslení ekvidistantní plochy k ploše f

```
ekvi:=plot3d(eq,0.5..2,3..4,numpoints=200,style=LINE);
```

definice jednotlivých snímků animace

```
frame[ii]:=display(
```

```
S(T,blue,2,['xG','yG','zG']),
S(TGU[1],black,2,['xU','yU','zU']),
S(TGL,red,2,['xL','yL','zL']),
Ob(TGL,2,1,red,2),
S(TGA[1],green,2,['xA','yA','zA']),
S(TGB[1],cyan,2,['xB','yB','zB']),
S(TGC[1],magenta,2,['xC','yC','zC']),
S(TGD[1],orange,2,['xD','yD','zD']),
S(TGE[1],black,1,['xE','yE','zE']),
kolo[1],
S(TGU[2],black,2,['xU','yU','zU']),
S(TGA[2],green,2,['xA','yA','zA']),
S(TGB[2],cyan,2,['xB','yB','zB']),
S(TGC[2],magenta,2,['xC','yC','zC']),
S(TGD[2],orange,2,['xD','yD','zD']),
S(TGE[2],black,1,['xE','yE','zE']),
kolo[2],
S(TGU[3],black,2,['xU','yU','zU']),
```

```

S(TGA[3],green,2,['xA`,`yA`,`zA`]),
S(TGB[3],cyan,2,['xB`,`yB`,`zB`]),
S(TGC[3],magenta,2,['xC`,`yC`,`zC`]),
S(TGD[3],orange,2,['xD`,`yD`,`zD`]),
S(TGE[3],black,1,['xE`,`yE`,`zE`]),
kolo[3],
S(TGU[4],black,2,['xU`,`yU`,`zU`]),
S(TGA[4],green,2,['xA`,`yA`,`zA`]),
S(TGB[4],cyan,2,['xB`,`yB`,`zB`]),
S(TGC[4],magenta,2,['xC`,`yC`,`zC`]),
S(TGD[4],orange,2,['xD`,`yD`,`zD`]),
S(TGE[4],black,1,['xE`,`yE`,`zE`]),
kolo[4],
end do:

```

Vykreslení výsledné animace

```
display(frame[d]$d=1..ii-1,insequence=true,scaling=constrained,orientation=[164,61]);
```